University of Amsterdam

Research Project 1

Generating probable password candidates for the assessment of Dutch domain password hashes

February 8, 2020

Student:
Tom Broumels
12632139

Supervisor:
ing. P. Campers

Assessor:
Prof. dr. ir. C.T.A.M. de Laat

Abstract

Human generated password authentication is commonly used today. Although hashes of passwords are stored to prevent the retrieval of plain text passwords, malicious actors can exploit predictability of human behaviour by attacking password hashes by utilising publicly available plain text passwords.

Earlier research has shown that passwords of different internet domains reflect language and culture related to these domains. This research explored the effect of using known Dutch domain related passwords for assessing Dutch domain NT Lan Manager (NTLM) hashes that meet Windows password complexity requirements.

The research question answered is "How do different password guessing algorithms compare in selecting password candidates for the assessment of NTLM password hashes for Dutch domain users?". To answer this question, different approaches have been compared in both amount of retrieved hashes within 30 minutes and amount of hashes found in total on two sets of Dutch domain NTLM hashes. The approaches used are Probabilistic Context Free Grammar (based on the work of Weir et al.), Generative Adversarial Network (based on the work of Hitaj et al.) and dictionaries based on Dutch domain passwords and Dutch words.

This report shows that approaches based on Dutch domain passwords increase the amount of passwords found within 30 minutes compared to internationally oriented dictionaries. In particular when using the Probabilistic Context Free Grammar.

This report also shows that combining Dutch domain password based approaches with internationally oriented dictionary Weakpass_2 can lead to an approach that finds both considerably more passwords within a 30 minute interval and more passwords in total.

1 Introduction

Although password authentication is not considered to be the most secure authentication method, it still is a widely used option in practice today, mainly because of usability and deployability characteristics [1].

From early on, password authentication has been the target of attacks [2]. As a result, techniques and procedures related to password authentication have been improved, e.g.:

- Efficient attacks using rainbow tables have been introduced to enable pre-computed hash lookups [3]. To mitigate such attacks, amongst others, password policies and salts have been used.
- Graphics processing units (GPUs) are being utilized for checking large amounts of password candidates per second. To counter such attacks, computationally expensive and memory intensive hashing algorithms have been developed [4].

Nowadays, a fair amount of research related to attacking passwords focuses on developing approaches for selecting likely password candidates. This includes successfully utilizing breached passwords to crack considerable amounts of hashes [5, 6, 7], and using publicly available information of persons and organisations [8, 9]. From a defensive point of view, similar algorithms are being utilized to score the strength of newly selected passwords since determining the entropy of user generated passwords is not a trivial task [10, 11].

Some studies compare passwords of different country domains. These studies indicate that user selected passwords reflect user language or cultural user aspects [5, 6, 12]. Such information could offer opportunities for more efficiently assessing Dutch domain passwords.

Our research focused on assessing the strength of passwords of Dutch users by taking known breach corpus data of Dutch domain users, i.e. users with ".nl" email addresses, as a starting point. The results can support security assessments, e.g. red teaming exercises, and support the further development of preventive measures to assure stronger password selection for Dutch domain services.

The research is commissioned by Secura, a Dutch company specialised in digital security.

2 Ethical considerations

Publicly available breach corpuses are commonly used for password related research [5, 6, 7, 9]. Furthermore, guidelines, e.g., the National Institute of Standards and Technology (NIST) Digital Identity Guidelines explicitly suggest the usage of password corpuses for assessing password strength [13].

3 Context

To be able to utilize the results of this research at Secura, we have narrowed down the scope based on conditions commonly seen in practice. We looked into two use cases for cracking hashes:

- Short term cracking, where retrieval of passwords is desired within 30 minutes.
- Extended cracking, where retrieval of a larger number of passwords in total is important. These assessments are often ran overnight.

We focused on assessing NT Lan Manager (NTLM) password hashes for Dutch domain users and assumed that the users related to the hashes are not always known.

Typically, the organisations where the hashes belong to have enabled Microsoft password complexity requirements. These requirements force users to, amongst others, choose passwords of a minimum length of 8 characters and use at least three of the following character types: uppercase characters, lowercase characters, digits, symbols and special characters. We will refer to passwords matching password complexity requirements as "8ULNS passwords" in this report from now on.

In our research we omitted approaches that assume the use of unsalted hashes, e.g. rainbow tables, because the amount of assessed salted hashes is expected to increase in the near future.

4 Research question

Based on the goal and context of the research, the following research question was formulated:

"How do different password guessing algorithms compare in selecting password candidates for the assessment of NTLM password hashes for Dutch domain users?"

5 Related work

This section describes approaches for password candidate selection followed by research currently available on Dutch passwords.

5.1 Prior work on password candidate selection

In 2005, Narayanan and Shmatikov [14] described approaches based on Markov modeling, utilizing the distribution of characters in the native language of users to reduce the key space being searched by skipping unlikely password candidates. This enabled them to generate longer probable password candidates within a reasonable amount of time. This approach was improved by M. Dürmuth et al. [9] in 2013 by, among others, introducing Ordered Markov Enumerator (OMEN).

In 2006, C. Kuo et al. [15] researched attacks on mnemonic phrase-based passwords, and in 2016 K. Young [16] presented a way to retrieve passphrases using various online sources such as Wikipedia texts.

In 2009, Weir et al. [7] described Probabilistic Context-Free Grammars (PCFG). It is based on building a grammer by splitting known passwords up into parts (i.e. alphabetic characters, digits and special characters). The grammar is used for generating new password candidates from those parts in such a way that likely combinations are generated first.

In 2016, Ding Wang et al. [8] created a framework called TarGuess for online targeted guessing using a combination of algorithms based on PCFG, Markov and Bayesian theory, gaining success rates up to 32% against users of security related forums and up to 73% against other users within 100 guesses per user. The researchers approach password cracking by combining different techniques and algorithms.

In 2019, Hitaj et al. [17] demonstrated the use of deep learning, i.e. a Generative Adversarial Network (GAN), for generating password candidates based on known passwords, claiming to be able to generate candidates that are not part of the training set and are unlikely to be generated by using commonly used rulesets or Markov-based algorithms. In addition, combining the GAN with word mangling rules yielded better results.

In 2020, Dutch security researchers Gevers and van Beek, experienced in large scale cracking at ScatteredSecrets.com, recognised that the approach of a cracking job starts with defining the goal of the job and determining the justified investment for reaching that goal in terms of cracking time and costs (e.g. hardware, power). Depending on the amount of hashes and the type of hashes (e.g. MD5/Bcrypt, salted/unsalted), a feasible attack could be defined as selecting a key space that can be searched without exceeding the intended investment.

From 2013 on, Hunt published an online service, HaveIBeenPwned.com, that enables users to check known password breaches for the existence of leaked passwords related to email addresses. Hunt has been known to check validity of breached hashes and passwords to some extend before publishing the hashes of passwords in so called Pwned password lists.

The work of Chaabane et al. (2012) and Jaeger et al. (2016) identified categories of subjects where passwords commonly refer to while attacking passwords [18, 6]. Brown et al. (2014) mention categories of subjects where passwords of US students refer to based on interviews with these students [19].

5.2 Prior work on domain specific passwords

In opposition to the large amount of research on international passwords, only limited research is available that focuses on the analysis of Dutch domain passwords, more specifically: passwords selected by users of web based Dutch .nl services.

In 2008 van Heerden and Vorster [12] presented Markov models for English, Dutch and Swahili passwords and briefly noted that assigned probabilities between characters in the generated models are different for different languages.

In 2010 Dell'Amico et al. [5] have shown that using dictionaries related to the language of its users, i.e. Italian and Finnish, increases the number of retrieved passwords.

In 2016 Jaeger et al. [6] have identified common passwords per country related domain by analysing breached password corpuses. They found that popular passwords for a country related domain can be specific to the language of users.

Recently, August 2019, the Dutch newspaper Het Financieele Dagblad published [20] an article on frequently selected passwords by Dutch users. The article was based on the work of Dutch security researchers Gevers and van Beek. They found that the most frequently used passwords consist of a considerable amount of Dutch words and (references to) Dutch names such as first names and names of soccer teams.

6 Methodology

To answer the main research question, we have divided our research into four phases. We started by preparing Dutch domain password data that we would use as input for different password cracking approaches. After that, we selected a limited amount of password cracking approaches for comparison. Using a lab experiment, we compared these approaches and analysed (combinations of) approaches before proposing a suitable approach for the use cases under consideration.

6.1 Dutch domain password selection and cleaning

We started our research by extracting all Dutch domain plain text passwords from breach corpuses, cleaning the data and analysing the characteristics of the data. This process is visualized in image 1.

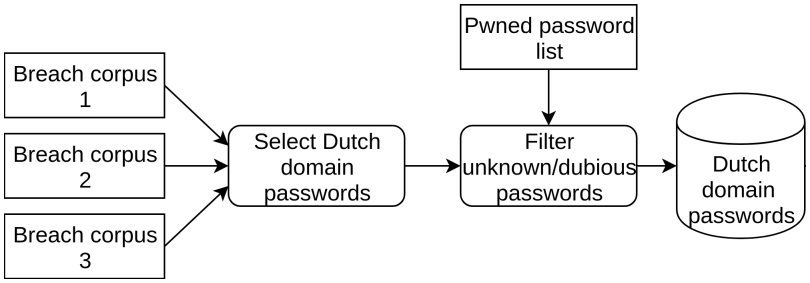

Figure 1: Data selection and cleaning process.

Three commonly mentioned breach corpuses have been used (listed in "Appendix A: breach corpuses"). Each breach corpus exists of text files containing an email/password pair separated by a colon character on each line. All entries starting with a Dutch domain email address, i.e. email addresses ending with ".nl", followed by a colon have been selected as potential Dutch domain entries.

Because the origin of the breach corpuses is unclear, we removed all the entries that contained a hash that was not listed in Troy Hunts Pwned password list version 5.

The remaining list still contained passwords that did not seem to represent user generated passwords. We removed those entries that had one of the following properties:

- Passwords that match patterns of common hashes, e.g., "PBKDF1:sha1". We assume that these entries are the result of cracking the wrong password type, e.g., cracking sha1($pass) instead of sha1(sha1($pass))
- Passwords containing email addresses followed by a colon. We assume that these entries are the result of incorrect processing of email/password entries during the creation of the breach corpus.

The union of the remaining entries are used for the remainder of this research project. A brief analysis of the Dutch domain passwords was performed.

6.2 Selection of password cracking approaches

For measuring the performance of approaches on Dutch domain passwords, we selected known approaches that (indirectly) take known passwords as input. Other selection criteria were difference in algorithm, and the availability of software to generate password candidates.

We identified three categories for selecting password candidates: (re)combining substrings of known passwords (e.g., PCFG, OMEN, PRINCE, etc.), machine learning based on known passwords (Neural Network, GAN), and approaches based on human behaviour (specific word lists, Dutch words, mnemonic based cracking, etc.). We have selected at least one approach from each category:

- Probabilistic Context Free Grammar (PCFG), based on the work of Weir et al. [7]. Candidate generation relies on splitting up known passwords into parts (i.e. alphabetic, digits and special characters) and deriving rules for making combinations of these parts and recombining common parts first. Common parts are determined by assessing frequencies in the passwords used as input for the algorithm. Existing PCFG Cracker software has been used [21], the model is trained using all Dutch domain passwords including duplicates as required for training a well performing model.
- Generative Adversarial Network (GAN), based on the work of Hitaj et al. [17]. Candidate generation is based on training two deep learning models: a generator and a discriminator. First, the discriminator model learns to distinguish real Dutch domain passwords from random input. After that, the generator model is trained to generate password candidates that are recognised by the discriminator as passwords using feedback of the discriminator. Existing PassGAN software has been used [22]. The recommended 200,000 iterations have been used for training.
- Dutch domain passwords Dictionary, containing the unique Dutch domain passwords found in the password corpuses. We will refer to this approach in the results section as NLPASS.
- Lists Dictionary, a collection of words related to categories that are similar to categories mentioned in the work of Chaabane et al.[19], Jaeger et al.[18] and Brown et al.[6]. Some additional lists of words are added for categories identified by manually analysing the Dutch domain passwords not yet covered. The categories used can be found in "Appendix D: Dutch domain baseword analysis". We will refer to this approach in the results section as NLLISTS.
- Dutch words dictionary, containing words found in dictionaries, Dutch Wikipedia articles and Dutch news articles. We will refer to this approach in the results section as NLWORDS.
- A combination of NLLISTS, NLPASS and NLWORDS, since NIST suggests to check for, amongst others, a combination of this data. We will refer to this approach in the results section as NLCOMBO.

6.3 Measuring performance of approaches

We were interested in the total amount of hashes cracked as well as the point in time that hashes were cracked. We have designed an experiment that enables us to register the hashes and timestamp of cracking for each test thereby enabling us to plot the metrics in graphs. Additionally, these measurements enabled us to get insight in the expected performance of combined approaches.

We added additional internationally oriented dictionaries to get an insight in how internationally oriented approaches perform compared to Dutch domain based approaches. Also we repeated the measurements using different word mangling rules. All variables used in the experiment are visualised in Figure 2

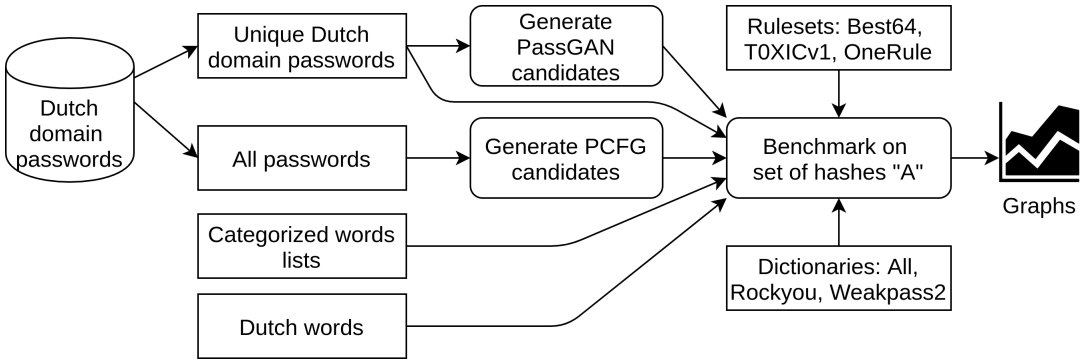

Figure 2: Variables used in the designed experiment.

The hashes used for our experiment are hashes of enabled Active Directory users for a Dutch company. The hashes were collected in December 2019 and were not known to be breached. The Active Directory password policy used is 8ULNS. We will refer to this set of hashes as set "A" in this report.

Word mangling rules are commonly used in conjunction with password candidates to increase the key space searched by generating additional (slightly) different password candidates based on the password candidate in the dictionary. For our experiment we used: no word mangling rules, Best64 (Hashcat 3.6.0 version), T0XICv1 (Hashcat 3.6.0 version) and OneRuleToRuleThemAll [23].

To have a reference while interpreting the results of the approaches based on Dutch passwords, we also include two commonly used internationally oriented dictionaries in our experiment: RockYou (14,344,390 entries) and Weakpass2 (2,649,982,129 entries) [24, 25]. Also a dictionary created by Secura named All (4,103,276,873 entries) is included in the experiment. This dictionary is an ordered list containing the union of entries of several dictionaries and word lists. The PCFG and GAN models trained were able to generate billions of password candidates. For practical reason we decided to generate dictionaries containing the same amount of password candidates as the Weakpass_2 dictionary (2,649,982,129 entries).

All tests are performed on a dedicated cracking system running Hashcat 3.6.0. A new Hashcat pot file is created to make sure hashes that have been cracked and stored during earlier test will not be used for fast lookups in later tests. For each test a GTX 1080 Ti GPU is used for cracking.

Since we selected the measured approaches based on their differences, we expected them to generate different password candidates in different moments in time. This would mean that combining approaches could lead to an increase in cracked hashes. To validate this, we calculated the amount of hashes that would be cracked by combined approaches by counting the unique entries in the union of the cracked hashes.

6.4 Proposing a well performing approach

Based on the results of the experiment, we selected a combination of four different approaches that performed well on set "A". We included approaches with steep curves, omitted very shallow curves and removed final parts of dictionaries that contained very unlikely password candidates. This resulted in the selection of a combination of the following dictionaries:

- The NLCOMBO dictionary found a relatively large amount of passwords considering the limited size of the dictionary. For that reason it was added entirely at the start of the new dictionary.
- Because the PCFG approach generates more likely candidates first, the amount of found passwords decreases soon. For that reason only the first 75% of the rows of the PCFG dictionary were added to the new NLWOVEN dictionary.
- As the GAN approach generates more password candidates, it becomes less likely that new passwords are being generated. For that reason, only the first 50% of the rows of the GAN dictionary are being used.

- Measurements of the Weakpass_2 dictionary on "A" showed a more shallow curve and still retrieved passwords while nearing the end of the dictionary. Therefore, the entire Weakpass_2 dictionary was included.

The lines of the dictionaries of these approaches were woven together into a new dictionary based on the plotted graphs for the benchmarks on "A". More lines were added for dictionaries with steeper curves up to the point that the curves became shallow, and by adding less lines from that point on until reaching the end of the dictionary.

Before combining the (partial) dictionaries, duplicate entries between dictionaries were removed in the dictionary with the most shallow curve in the start of its graph of plotted results. In total 9.1% of all lines were removed.

To validate the proposed approach NLWOVEN, we repeated the experiment for all approaches on a different set of hashes of a different Dutch organisation that also enforces 8ULNS passwords. We will refer to that set of hashes as set "B".

7 Results

The next subsections list the results for the collection and analysis of password data, and the results for the performed experiment.

7.1 Dutch domain password data selection and cleaning

Figure 3 shows a Venn diagram of the amount of Dutch domain passwords present in the examined breach corpuses after cleaning. "C1", "DP" and "BC" refer to the names of the breaches listed in "Appendix A: breach corpuses". In the cleaning process, 176,956 unique passwords (used in 190,152 email/password entries) have been removed because the hashes of these passwords were not listed in the list of known password hashes that are published by Hunt in Pwned passwords list V5. Afterwards, another 20,488 passwords (used in 24,406 email/password combinations) have been removed that looked like password hashes.
Based on the remaining entries, two lists of Dutch domain passwords have been generated: one list containing 2,355,739 rows representing all unique passwords, and another list containing the password for each unique username/password combination (3,424,464 rows in total). This means that 31% of the Dutch domain passwords are being used multiple times, e.g. being re-used by the same user for another email address, or by other people who selected the same password.

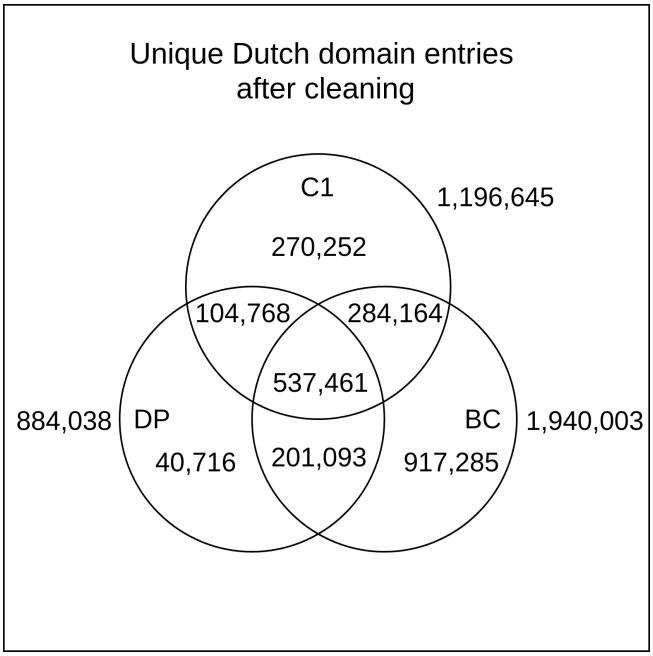

Figure 3: Unique Dutch domain passwords. Amount of entries per breach corpus are mentioned at the side.

The ten most common Dutch domain passwords and Dutch domain 8ULNS passwords found after combining and cleaning the breach compilations are listed in tables 1 and 2 respectively. Counted frequencies and percentage of all Dutch domain related passwords are mentioned. Extended lists containing the top 50 most common passwords can be found in "Appendix B: Top 50 Dutch domain passwords". Additional statistics on Dutch domain passwords are listed in "Appendix C: Dutch domain password properties".

Password	Frequency	Percentage of total
123456	8795	0.26
welkom	3950	0.12
SKIFFY	3708	0.11
welkom1	2547	0.07
123456789	2524	0.07
qwerty	2304	0.07
welkom01	2220	0.06
wachtwoord	2177	0.06
geheim	1792	0.05
amsterdam	1568	0.05

Table 1: Top 10 Dutch domain breach compilation passwords.

Password	Frequency	Percentage of total
Welkom01	861	0.26
ka_dJKHJsy6	198	0.06
Welkom123	187	0.06
PPPrt30TA	152	0.05
Feyenoord1	139	0.04
P@ssw0rd	107	0.03
Amsterdam1	102	0.03
Hallo123	101	0.03
Wachtwoord1	94	0.03
Geheim01	76	0.02

Table 2: Top 10 8ULNS Dutch domain breach compilation passwords.

7.2 Performance of (combinations of) approaches

The results for OneRule are consistently outperforming other rulesets or the use of no rulesets at all. For that reason, this section only shows the results for OneRule. A complete overview of results is available in "Appendix E: Additional experimental results".

Table 3 lists an overview of the approaches used in the experiment.

Approach	Name	Type
GAN	Generative Adversarial Network	Machine learning
PCFG	Probabilistic Context-Free Grammar	Frequencies
NLLISTS	Categorical words	Human behaviour
NLWORDS	Dutch words	Dutch dictionary
NLPASS	Dutch domain passwords	Breach corpuses
NLCOMBO	Combination of NLLISTS, NLWORDS and NLPASS	Combination of approaches
ROCKYOU	Internationally oriented dictionary Rockyou	Dictionary
WEAKPASS2	Internationally oriented dictionary Weakpass_2	Dictionary
ALL	Dictionary created by Secura	Compilation of dictionaries
NLWOVEN	Combination of GAN, PCFG, NLCOMBO and WEAKPASS2	Combination of approaches

Table 3: Overview of the compared approaches.

Table 4 and 5 show the percentage of cracked passwords after 30 minutes and the percentage of cracked passwords after exhausting the full dictionary, respectively.

Hashes	Hash type	Unique hashes	Ruleset	GAN	PCFG	NLLISTS	NLWORDS	NLPASS	NLCOMBO	ROCKYOU	WEAK-PASS2	ALL	NLWOVEN
A	NTLM	1338	OneRule	31.46%	52.32%	11.51%	35.28%	35.80%	41.63%	41.70%	39.09%	25.71%	55.75%
B	NTLM	1702	OneRule	15.69%	30.79%	4.00%	12.69%	12.81%	16.80%	19.92%	27.26%	7.34%	32.90%

Table 4: Percentage of hashes cracked after 30 minutes. Please note that a considerable amount of approaches already finished before the 30 minute limit.

Hashes	Hash type	Unique hashes	Ruleset	GAN	PCFG	NLLISTS	NLWORDS	NLPASS	NLCOMBO	ROCKYOU	WEAK-PASS2	ALL	NLWOVEN
A	NTLM	1338	OneRule	39.46%	58.30%	11.51%	35.28%	35.80%	41.63%	41.70%	64.80%	65.40%	70.10%
B	NTLM	1702	OneRule	22.86%	40.25%	4.00%	12.69%	12.81%	16.80%	19.92%	50.24%	50.12%	53.53%

Table 5: Percentage of hashes cracked after exhausting full dictionaries.

The performance of combinations of dictionaries for "A" is shown in detail in Table 6. Approaches included in NLCOMBO are omitted in the overview since these would not yield better results than NLCOMBO would.

Table 7 shows the maximum percentage of cracked passwords that can be reached by combining different approaches.

	WEAK-PASS2	ROCKYOU	NLCOMBO	GAN	PCFG	ALL
WEAKPASS2	64.80%	64.80%	64.8%	66.89%	67.49%	65.84%
ROCKYOU	64.80%	41.70%	47.46%	52.47%	59.19%	65.40%
NLCOMBO	64.80%	47.46%	41.63%	53.14%	60.61%	65.40%
GAN	66.89%	52.47%	53.14%	39.46%	60.69%	67.26%
PCFG	67.49%	59.19%	60.61%	60.69%	58.30%	67.86%
ALL	65.84%	65.40%	65.40%	67.26%	67.86%	65.40%

Table 6: Maximum percentage of retrieved hashes of dataset "A"by combining two approaches.

Hashes	Hash type	Unique hashes	Ruleset	Max. 1	Max. 2	Max. 3	Max. 4	Max. 5
A	NTLM	1338	OneRule	65.4%	69.1%	70.0%	70.3%	70.6%
B	NTLM	1702	OneRule	50.2%	52.9%	53.7%	54.3%	54.8%

Table 7: Maximum percentage of retrieved hashes when using the best possible combination of one or more approaches before introducing NLWOVEN.

Figure 4 shows the results for the experiment set "A" and "B" for word mangling ruleset OneRule.

To clearly visualise the results, the approaches have been plotted in different graphs based on the runtime of the approaches. Similar graphs for the performance of other word mangling rulesets can be found in "Appendix E: Additional experimental results".

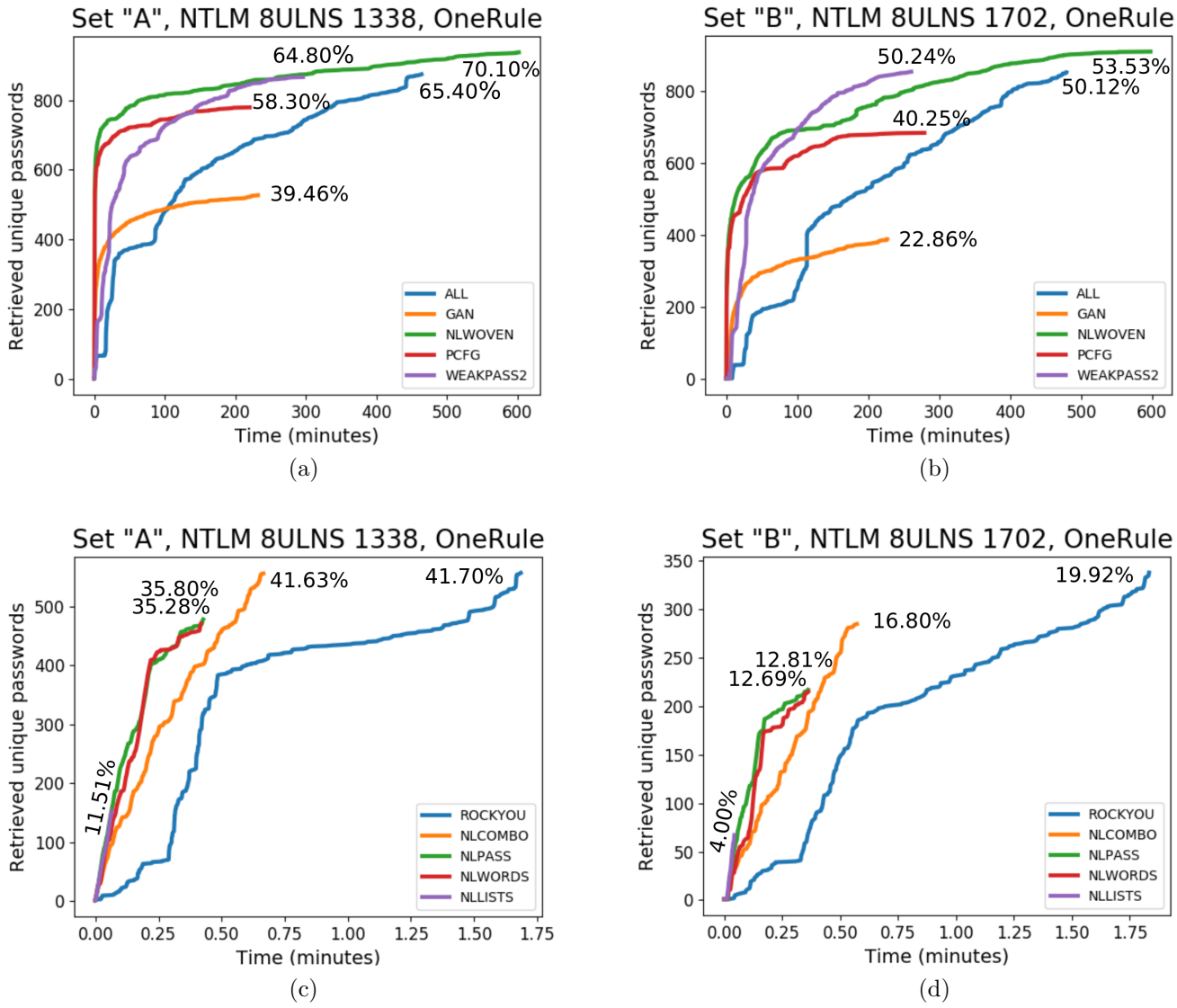

Figure 4: Cracked hashes for exhausted dictionaries of: "A" (left) and "B" (right), large dictionaries (top) and small dictionaries (bottom).

8 Discussion

In this section we discuss and interpret our results. We were interested in how (combined) approaches compare in terms of retrieved hashes within a 30 minute time frame and after exhausting the full dictionaries.

The OneRule word mangling ruleset increased the amount of found passwords considerably for all tests without increasing the time required for jobs to complete to an unacceptable duration. For that reason, we will limit the discussion to the results measured using OneRule.

8.1 Retrieved passwords within 30 minutes

As we can see in Figure 4 and Table 14, all five approaches based on the Dutch domain passwords have a considerably steeper curve early in the cracking process than the internationally oriented dictionaries. This is what one could expect since the Dutch language is, from an international perspective, not a commonly used language.

The small dictionaries NLLISTS, NLPASS and NLWORDS are exhausted within one minute, cracking between 11.51% and 35.80% of the hashes for "A" and between 4.00% and 12.81% for "B". While these values are not as high as the results for larger dictionaries, the amount of cracked hashes related to the amount of dictionary entries processed is high.

PCFG outperformed all compared approaches, including Weakpass_2, RockYou and All, with at least 10.62% for "A" and at least 3.53% for "B". GAN performs in a mediocre way compared to the other approaches and dictionaries on both sets.

8.2 Retrieved passwords after exhausting dictionaries

Figure 4 and Table 15 show that the approaches based on Dutch domain passwords crack less hashes compared to the international oriented dictionaries Weakpass_2 and All. Besides dictionary All containing more entries, another explanation for this difference could be that some Dutch domain users do use English words in passwords. We have confirmed that this is indeed the case by looking at the Dutch domain passwords (e.g., passwords contain "sunshine" as a base word).

While the difference in the amount of cracked passwords for PCFG, Weakpass_2 and All differ at most 7.10% for "A" and 9.99% for "B", GAN cracks considerably less hashes. This makes GAN an inefficient approach in terms of both cracked hashes in total and amount of processed entries per cracked hash.

8.3 Retrieved passwords for combined approaches

In Table 6 we can see the effect of combining two approaches. Some combinations lead to no increase at all, e.g. Weakpass_2 with RockYou, where other combinations lead to a larger amount of hashes cracked, e.g. an additional 2.69% for PCFG with Weakpass_2. Moreover, approaches individually not performing particularly well can lead to a higher total percentage when combined with an individual approach that already cracks a reasonable amount of hashes, e.g. GAN combined with PCFG leads to an increase of 2.39%. This indicates that such approaches, at least to some extent, search a different part of the key space.

Table 18 shows the maximum amount of hashes that can be cracked by combining approaches. Up to a combination of 4 approaches, it is possible to add another approach to the combination to increase the total amount of hashes cracked. The amount of dictionary entries that have to be processed however, raises in a more or less linear order where the increase in additional cracked passwords gets, in most cases, smaller for each additional approach added.

Figure 4 shows the results of the selected well performing approach NLWOVEN, based on a combination of GAN, PCFG, NLCOMBO and Weakpass_2. Because this approach is based on the results of set "A", one would expect NLWOVEN to outperform all other approaches after exhausting the entire dictionary on "A". It did so by at least 4.70% (a relative 7.19%). Additionally, it outperformed the best individual approach in the short term, PCFG, with 2,11% .

Validating the approach NLWOVEN on set "B" yields similar results: NLWOVEN outperformed the other approaches by cracking at least 3.29% (a relative 6,80%) more hashes after exhausting the entire dictionary. Within 30 minutes, all other approaches are outperformed with at least 2.11% additional hashes. At some point, Weakpass_2 outperforms NLWOVEN for some time. For the performance on "B" however, this is not influencing the results for the use cases under consideration.

If we compare the best previously used combination for finding passwords in total, All and T0XICv1, with the new combination of NLWOVEN and OneRule, we see that the retrieved passwords for "A" have increased from 58.37% to 70.10% (a relative 20,10%) and for "B" from 38.27% to 53.53% (a relative 38,25%), while retrieving considerably more passwords within 30 minutes: for "A" an increase from 24.96% to 55.75% (a relative 123,60%) and for "B" from 5.11% to 32.90% (a relative 543.84%).

8.4 Limitations

There are some limitations to our research. The proposed approach NLWOVEN has been validated on one set of 8ULNS hashes other than the set of hashes used for designing the approach. Performing tests on additional sets of hashes is required to be able to make statements about the performance of NLWOVEN on sets of 8ULNS hashes in general and to be able to optimize the way dictionaries are combined.

If we look at the extracted and cleaned Dutch domain passwords, we noticed passwords that are standing out, e.g. uppercase and non-Dutch word "SKIFFY", or probably fake passwords (e.g., "PPPrt30TA", because most username parts of the email addresses related to this password are only one or two characters long). For this reason, caution should be exercised when using statistics based on the used Dutch domain passwords.

For this study we based the categories included mainly on psychological research of US students. It is possible that a similar study under Dutch employees of Dutch organisations leads to different insights on the password selection behaviour of these users. Also we assumed that Dutch domain users select passwords in a similar fashion for public domain services as for accounts that are part of Active Directory.

Furthermore, we have only compared the Dutch domain based approaches to internationally oriented dictionaries and not to PCFG and GAN models trained on international breach corpus data.

9 Conclusions

The research started with the research question: *"How do different password guessing algorithms compare in selecting probable password candidates for assessing NTLM password hashes for Dutch domain users?"*

Our experiment on two sets of 8ULNS Dutch domain hashes has shown that using password cracking approaches based on breached Dutch domain passwords can retrieve passwords earlier in the cracking process than internationally oriented dictionaries Weakpass_2, RockYou and All. Large internationally oriented dictionaries however, are able to retrieve more passwords than individual Dutch domain password based approaches.

We have also shown that it is possible to combine different Dutch domain password based approaches with an internationally oriented dictionary to get an approach that retrieves more Dutch domain passwords in an early stage and will also crack more Dutch domain passwords in complete runs compared to using individual approaches or dictionaries.

Finally we report that using word mangling rules in addition to all the used cracking approaches had a considerable positive impact on the amount of retrieved Dutch domain passwords.

10 Future work

Comparing additional approaches, e.g. mnemonic phrase-based, can possible lead to more improved combinations of approaches.

More in depth analysis of Dutch domain passwords and the behaviour of Dutch users while selecting passwords could offer possibilities for retrieving more password hashes.

Previous work and further inquiry with experts in the field of password cracking learned that using retrieved passwords for a target organisation as input for a next run on the not yet cracked hashes, a so called "feedback loop", can result in cracking additional hashes.

Finally, information specific to the targeted organisation could be added for cracking, as suggested by NIST.

11 Acknowledgements

The author wishes to thank Secura: R. Moonen for the opportunity to do this research, P. Campers and E. Slangen for operational support and C. Hillen for reviewing research and documentation.

References

[1] Joseph Bonneau et al. "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes". In: *2012 IEEE Symposium on Security and Privacy*. IEEE. 2012, pp. 553–567.

[2] Robert Morris and Ken Thompson. "Password security: A case history". In: *Communications of the ACM* 22.11 (1979), pp. 594–597.

[3] Philippe Oechslin. "Making a faster cryptanalytic time-memory trade-off". In: *Annual International Cryptology Conference*. Springer. 2003, pp. 617–630.

[4] Alex Biryukov, Daniel Dinu, and Dmitry Khovratovich. "Argon2: new generation of memory-hard functions for password hashing and other applications". In: *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2016, pp. 292–302.

[5] Matteo Dell'Amico, Pietro Michiardi, and Yves Roudier. "Password strength: An empirical analysis". In: *2010 Proceedings IEEE INFOCOM*. IEEE. 2010, pp. 1–9.

[6] David Jaeger et al. "Analysis of publicly leaked credentials and the long story of password (re-) use". In: *International Conference on Passwords*. 2016.

[7] Matt Weir et al. "Password cracking using probabilistic context-free grammars". In: *2009 30th IEEE Symposium on Security and Privacy*. IEEE. 2009, pp. 391–405.

[8] Ding Wang et al. "Targeted online password guessing: An underestimated threat". In: *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. ACM. 2016, pp. 1242–1254.

[9] Claude Castelluccia et al. "When privacy meets security: Leveraging personal information for password cracking". In: *arXiv preprint arXiv:1304.6584* (2013).

[10] Matteo Dell'Amico and Maurizio Filippone. "Monte Carlo strength evaluation: Fast and reliable password checking". In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2015, pp. 158–169.

[11] Paul A Grassi et al. "NIST Special Publication 800-63b: Digital Identity Guidelines - Appendix A - Section A.1". In: *Enrollment and Identity Proofing Requirements. url: https://pages.nist.gov/800-63-3/sp800-63b.html* (2017).

[12] RP Van Heerden and JS Vorster. "Using Markov Models to crack passwords". In: *The 3rd International Conference on Information Warfare and Security: Peter Kiewit Institute, University of Nebraska, Omaha, USA*. 2008, pp. 24–25.

[13] Paul A Grassi et al. "NIST Special Publication 800-63b: Digital Identity Guidelines - section 5.1.1.2". In: *Enrollment and Identity Proofing Requirements. url:https://pages.nist.gov/800-63-3/sp800-63b.html* (2017).

[14] Arvind Narayanan and Vitaly Shmatikov. "Fast dictionary attacks on passwords using time-space tradeoff". In: *Proceedings of the 12th ACM conference on Computer and communications security*. ACM. 2005, pp. 364–372.

[15] Cynthia Kuo, Sasha Romanosky, and Lorrie Faith Cranor. "Human selection of mnemonic phrase-based passwords". In: *Proceedings of the second symposium on Usable privacy and security*. ACM. 2006, pp. 67–78.

[16] Hugo Labrande. *Crack Me I'm Famous: cracking weak passphrases using publicly-available sources.*

[17] Briland Hitaj et al. "Passgan: A deep learning approach for password guessing". In: *International Conference on Applied Cryptography and Network Security*. Springer. 2019, pp. 217–237.

[18] Abdelberi Chaabane, Gergely Acs, Mohamed Ali Kaafar, et al. "You are what you like! information leakage through users' interests". In: *Proceedings of the 19th Annual Network & Distributed System Security Symposium (NDSS)*. Citeseer. 2012.
[19] Alan S Brown et al. "Generating and remembering passwords". In: *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition* 18.6 (2004), pp. 641–651.
[20] Jan Fred van Wijnen. "Wachtwoord 'feyenoord' vaker gebruikt dan 'ajax', url:https://fd.nl/ondernemen/1310561/gekraakte-wachtwoorden-feyenoord-verslaat-ajax-met-grote-cijfers". In: *Het Financieele Dagblad (Dutch source)*. 2019.
[21] *PCFG cracker sourcecode on GitHub*. `https://github.com/lakiw/pcfg_cracker`. Accessed: 2019-01-20.
[22] *PassGAN sourcecode on GitHub*. `https://github.com/brannondorsey/PassGAN`. Accessed: 2019-01-20.
[23] *NotSoSecure OneRuleToRuleThemAll word mangling rules on GitHub*. `https://github.com/NotSoSecure/password_cracking_rules`. Accessed: 2019-01-20.
[24] *RockYou dictionary on weakpass.com*. `https://weakpass.com/wordlist/90`. Accessed: 2019-01-20.
[25] *Weakpass_2 dictionary on weakpass.com*. `https://weakpass.com/wordlist/1859`. Accessed: 2019-01-20.

Appendix A: breach corpuses

Online breached password corpuses used:

- BreachCompilation, abbreviated as "BC" in this report.
- Collection #1, abbreviated as "C1" in this report.
- Leaked Database Project, abbreviated as "DP" in this report.

Appendix B: Top 50 Dutch domain passwords

Dutch domain compilation (all passwords)

Password	Frequency	Percentage of total
123456	8795	0.26
welkom	3950	0.12
SKIFFY	3708	0.11
welkom1	2547	0.07
123456789	2524	0.07
qwerty	2304	0.07
welkom01	2220	0.06
wachtwoord	2177	0.06
geheim	1792	0.05
amsterdam	1568	0.05
151526	1373	0.04
feyenoord	1315	0.04
password	1226	0.04
12345	1175	0.03
lol123	1149	0.03
12345678	1098	0.03
voetbal	1042	0.03
rotterdam	979	0.03
linkedin	965	0.03
computer	959	0.03
willem	888	0.03
banaan	865	0.03
INH12345	863	0.03
Welkom01	861	0.03
vakantie	856	0.02
1234567	830	0.02
w8woord	817	0.02
jemoeder	807	0.02
000000	798	0.02
1234567890	750	0.02
1234	741	0.02
vergeten	732	0.02
mercedes	724	0.02
hoihoi	701	0.02
jeroen	696	0.02
pokemon	665	0.02
hallo	649	0.02
123123	640	0.02
hallo123	639	0.02
rakker	612	0.02
tijger	610	0.02
konijn	610	0.02
lollol	607	0.02
muziek	600	0.02
hoi123	586	0.02
koffie	565	0.02
dennis	558	0.02
dolfijn	554	0.02
qwerty123	554	0.02
internet	543	0.02

Dutch domain compilation (8ULNS passwords)

Password	Frequency	Percentage of total
Welkom01	861	0.26
ka_dJKHJsy6	198	0.06
Welkom123	187	0.06
PPPrt30TA	152	0.05
Feyenoord1	139	0.04
P@ssw0rd	107	0.03
Amsterdam1	102	0.03
Hallo123	101	0.03
Wachtwoord1	94	0.03
Geheim01	76	0.02
Trustno1	68	0.02
Tomaat95	67	0.02
Formule1	61	0.02
Voetbal1	61	0.02
Welkom02	59	0.02
Welkom12	58	0.02
Rotterdam1	57	0.02
Monique1	56	0.02
ikbg38_38	56	0.02
Qwerty123	56	0.02
Kikkervis7	55	0.02
Qwerty12	52	0.02
Linkedin1	51	0.02
Welkom2011	51	0.02
Passw0rd	50	0.02
Linkedin01	47	0.01
Amsterdam01	47	0.01
The8upUd	46	0.01
Maarten1	46	0.01
Zomer2011	45	0.01
Nederland1	44	0.01
Willem01	43	0.01
Thomas01	42	0.01
Utrecht1	42	0.01
Porsche911	40	0.01
Appeltaart1	40	0.01
Martijn1	40	0.01
Jolanda1	40	0.01
Computer1	39	0.01
Beertje1	39	0.01
W0rdp4ss	39	0.01
Raktak12	38	0.01
Icqfjm86	38	0.01
Kachien86	38	0.01
sliCknu3	37	0.01
SZ9kQcCTwY	36	0.01
Gangster2	35	0.01
Richard1	35	0.01
Marieke1	35	0.01
L1nk3d1n	35	0.01

Table 8: 50 most frequent passwords for all Dutch domain passwords (left) and 50 most frequent passwords of 8 or more characters containing at least one character of minimal 3 out of 4 character sets (lower case, upper case, digit, symbol). In total there are 3,424,464 Dutch domain passwords of which 328,984 are 8ULNS passwords.

Appendix C: Dutch domain password properties

Table 9 shows characteristics of the Dutch domain passwords used. In total there are 3,424,464 Dutch domain passwords of which 328,984 are 8ULNS passwords.

Characteristic	**% of Dutch domain passwords**	**% of Dutch domain 8ULNS passwords**	**Example**
Only lowercase alpha	1298805 (37.93%)	0 (0.00%)	aaaaaaaa, qrstuvwxyz
Only uppercase alpha	13995 (0.41%)	0 (0.00%)	AAAAAAAA, QRSTUVWXYZ
Only alpha	1312800 (38.34%)	0 (0.00%)	Password, MyPaSsWoRd
Only numeric	189708 (5.54%)	0 (0.00%)	aaaaaaaa, abcdabcd
First capital last symbol	11783 (0.34%)	9676 (2.94%)	Password!, Pwd@@@@@@@@
First capital last number	243167 (7.10%)	183643 (55.82%)	Password1, Welkom01
Single digit on the end	337480 (9.85%)	53678 (16.32%)	Password1, Abcdefghi1
Two digits on the end	554585 (16.19%)	88773 (26.98%)	Password01, Password02
Three digits on the end	190379 (5.56%)	26706 (8.12%)	Password123, Password456

Table 9: Character set usage for Dutch domain passwords

Tables 10, 11 and 12 show the digits used at the end of all Dutch domain passwords:

Last 3 digits	Count	Percentage of passwords
123	76730	2.24%
234	14161	0.41%
456	13121	0.38%
000	12656	0.37%
001	10928	0.32%
007	10649	0.31%
010	9376	0.27%
345	8704	0.25%
011	7997	0.23%
009	6357	0.19%

Table 10: Last three digits used in Dutch domain passwords

Last 4 digits	Count	Percentage of passwords
1234	13271	0.39%
3456	11301	0.33%
2345	7839	0.23%
2010	7305	0.21%
2011	6775	0.20%
2000	6690	0.20%
2009	5278	0.15%
2008	4858	0.14%
2007	4178	0.12%
1995	4041	0.12%

Table 11: Last four digits used in Dutch domain passwords

Last 5 digits	Count	Percentage of passwords
23456	11159	0.33%
12345	7667	0.22%
56789	3477	0.10%
45678	1415	0.04%
51526	1377	0.04%
23123	1345	0.04%
00000	1309	0.04%
34567	1279	0.04%
54321	1204	0.04%
67890	1064	0.03%

Table 12: Last five digits used in Dutch domain passwords

Appendix D: Dutch domain baseword analysis

To get an insight in words of which categories are used, we have matched the Dutch domain passwords case-insensitively with Dutch online wordlists related to Dutch subjects (e.g., names, hobbies, artists). Words containing special characters have been added twice: as is, and a copy without the special characters.

As we do not know the exact motivation of the person who selected the passwords, we would expect to find false positives, e.g., "Rotterdam" contains both the city "Rotterdam" as well as the animal "otter". And, e.g., "haanhenkip" contains references to animals but also the name "henk". For this reason, we aim to decrease the number of false positives by excluding a subset of passwords by using the following criteria for matching:

- Only words with a minimal length of three characters are used for matching
- A word matches if there are no characters in front of the word in the password that are part of the same character set as the first character as the word, e.g., "ajax4ever" and "!ajax!voetbal!" do match and "hupajax" does not match.
- A word matches if there are no characters after the word in the password that are part of the same character set as the last character as the word, e.g., "ajax10" and "ajax" do match and "ajaxvoetbal" does not match.

Table 12 lists the counted occurences of words for specfic categories.

Category	**Matching passwords**	**Matching unique passwords**	**Elements in wordlist**
First names	531337(15.52%)	267085(11.34%)	9348
Family names	203503(5.94%)	107539(4.56%)	9113
Pet names	159859(4.67%)	71978(3.06%)	646
Cities and townships	64118(1.87%)	30498(1.29%)	7120
Comic character names	43515(1.27%)	19593(0.83%)	774
Animals	32178(0.94%)	13445(0.57%)	4924
Payed soccer teams	28638(0.84%)	8130(0.35%)	310
Brands	17403(0.51%)	7039(0.30%)	254
Hobbies, sports and interests	17031(0.50%)	6128(0.26%)	496
Affix ("tussenvoegsel")	1280(0.04%)	928(0.04%)	22
Months and seasons	14338(0.42%)	6555(0.28%)	26
Religion	13141(0.38%)	6982(0.30%)	1211
Artists	12934(0.38%)	5924(0.25%)	2390
Amateur soccer teams	11413(0.33%)	6624(0.28%)	6009
Planets and zodiac signs	10040(0.29%)	4372(0.19%)	122
Jobs	9094(0.27%)	3879(0.16%)	1308
Countries	8713(0.25%)	3504(0.15%)	289
Breached organisations	8696(0.25%)	3793(0.16%)	552
Curse words	8058(0.24%)	3411(0.14%)	295
Keyboard walks	7020(0.20%)	2398(0.10%)	4688
Dutch canon subjects	6069(0.18%)	1776(0.08%)	148
Human character types	3035(0.09%)	2118(0.09%)	241
Provinces	2182(0.06%)	645(0.03%)	17
Dutch date pattern (dd-mm-yy or dd-mm-yyyy)	78285(2.29%)	55284(2.35%)	
Dutch zipcode pattern (4 digits+2 alphabetic characters)	3498(0.10%)	3250(0.14%)	
At least one exact match	1173866(34.28%)	705053(29.93%)	

Table 13: Base words found in Dutch domain passwords.

Appendix E: Additional experimental results

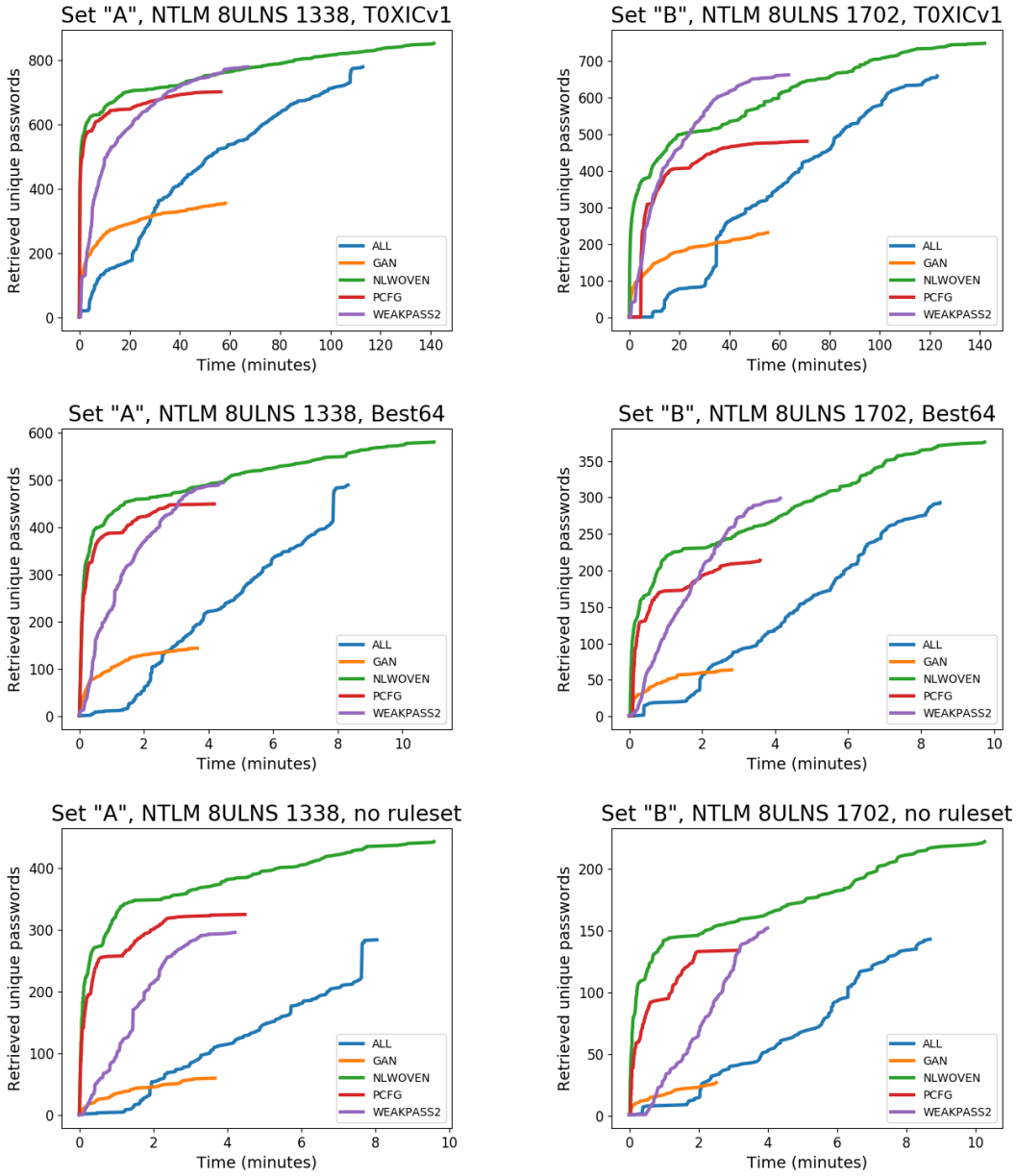

Figure 5: Plotted cracked hashes for large dictionaries on both sets of hashes: "A" (left) and "B" (right) for rulesets T0XIC (top), Best64 (middle) and no ruleset (bottom).

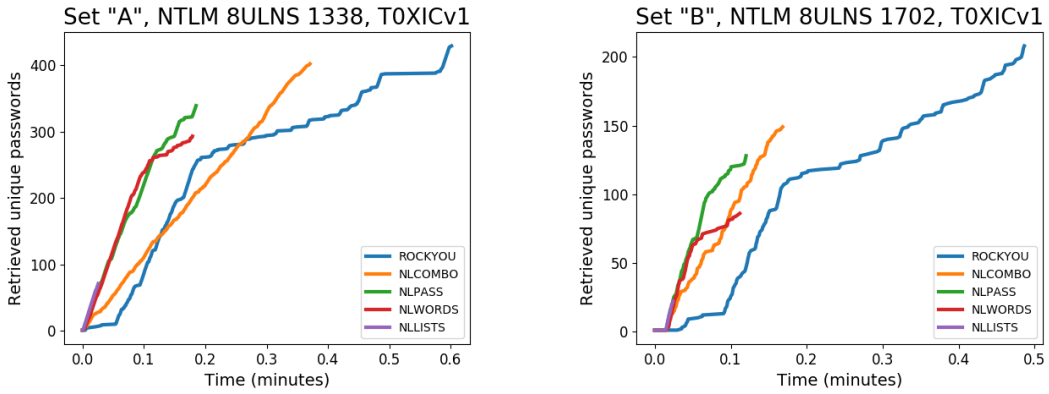

Figure 6: Plotted cracked hashes for small dictionaries on both sets of hashes: "A" (left) and "B" (right) for ruleset T0XICv1.

Hashes	Hash type	Unique hashes	Ruleset	GAN	PCFG	NLLISTS	NLWORDS	NLPASS	NLCOMBO	ROCKYOU	WEAKPASS2	ALL	NLWOVEN
A	NTLM	1338	(none)	4.56%	24.36%	0.00%	0.22%	3.06%	3.14%	3.06%	22.20%	21.30%	33.18%
A	NTLM	1338	Best64	10.84%	33.63%	1.64%	7.17%	9.87%	12.41%	12.03%	37.00%	36.62%	43.42%
A	NTLM	1338	T0XICv1	23.84%	50.45%	5.38%	21.97%	25.41%	30.12%	32.14%	49.93%	24.96%	53.29%
A	NTLM	1338	OneRule	31.46%	52.32%	11.51%	35.28%	35.80%	41.63%	41.70%	39.09%	25.71%	55.75%
B	NTLM	1702	(none)	1.65%	7.93%	0.06%	0.12%	0.76%	0.76%	0.94%	8.99%	8.46%	13.10%
B	NTLM	1702	Best64	3.82%	12.63%	0.41%	1.65%	2.35%	3.00%	4.00%	17.63%	17.27%	22.15%
B	NTLM	1702	T0XICv1	11.46%	25.79%	1.35%	5.11%	7.58%	8.81%	12.28%	32.96%	5.11%	29.91%
B	NTLM	1702	OneRule	15.69%	30.79%	4.00%	12.69%	12.81%	16.80%	19.92%	27.26%	7.34%	32.90%

Table 14: Percentage of hashes cracked after 30 minutes. Please note that approaches, i.e. small dictionaries, may have finished before the 30 minutes passed. limit.

Hashes	Hash type	Unique hashes	Ruleset	GAN	PCFG	NLLISTS	NLWORDS	NLPASS	NLCOMBO	ROCKYOU	WEAKPASS2	ALL	NLWOVEN
A	NTLM	1338	(none)	4.56%	24.36%	0.00%	0.22%	3.06%	3.14%	3.06%	22.20%	21.30%	33.18%
A	NTLM	1338	Best64	10.84%	33.63%	1.64%	7.17%	9.87%	12.41%	12.03%	37.00%	36.62%	43.42%
A	NTLM	1338	T0XICv1	26.61%	52.54%	5.38%	21.97%	25.41%	30.12%	32.14%	58.37%	58.37%	63.83%
A	NTLM	1338	OneRule	39.46%	58.30%	11.51%	35.28%	35.80%	41.63%	41.70%	64.80%	65.40%	70.10%
B	NTLM	1702	(none)	1.65%	7.93%	0.06%	0.12%	0.76%	0.76%	0.94%	8.99%	8.46%	13.10%
B	NTLM	1702	Best64	3.82%	12.63%	0.41%	1.65%	2.35%	3.00%	4.00%	17.63%	17.27%	22.15%
B	NTLM	1702	T0XICv1	13.63%	28.26%	1.35%	5.11%	7.58%	8.81%	12.28%	38.90%	38.72%	43.95%
B	NTLM	1702	OneRule	22.86%	40.25%	4.00%	12.69%	12.81%	16.80%	19.92%	50.24%	50.12%	53.53%

Table 15: Percentage of hashes cracked after exhausting full dictionaries.

Hashes	Hash type	Unique hashes	Ruleset	GAN	PCFG	NLLISTS	NLWORDS	NLPASS	NLCOMBO	ROCKYOU	WEAKPASS2	ALL	NLWOVEN
A	NTLM	1338	(none)	61	326	0	3	41	42	41	297	285	444
A	NTLM	1338	Best64	145	450	22	96	132	166	161	495	490	581
A	NTLM	1338	T0XICv1	356	703	72	294	340	403	430	781	781	854
A	NTLM	1338	OneRule	528	780	154	472	479	557	558	867	875	938
B	NTLM	1702	(none)	28	135	1	2	13	13	16	153	144	223
B	NTLM	1702	Best64	65	215	7	28	40	51	68	300	294	377
B	NTLM	1702	T0XICv1	232	481	23	87	129	150	209	662	659	748
B	NTLM	1702	OneRule	389	685	68	216	218	286	339	855	853	911

Table 16: Amount of cracked hashes per approach after exhausting full dictionaries.

	WEAKPASS2	ROCKYOU	NLCOMBO	GAN	PCFG	ALL
WEAKPASS2	50.24%	50.24%	50.24%	51.53%	52.12%	51.23%
ROCKYOU	50.24%	19.92%	22.62%	30.02%	41.36%	50.12%
NLCOMBO	50.24%	22.62%	16.80%	27.85%	40.54%	50.12%
GAN	51.53%	30.02%	27.85%	22.86%	42.30%	51.35%
PCFG	52.12%	41.36%	40.54%	42.30%	40.25%	52.12%
ALL	51.23%	50.12%	50.12%	51.35%	52.12%	50.12%

Table 17: Maximum percentage of retrieved hashes of dataset B when combinaing two approaches.

Hashes	Hash type	Unique hashes	Ruleset	Max. 1	Max. 2	Max. 3	Max. 4	Max. 5
A	NTLM	1338	(none)	24.4%	30.7%	33.1%	33.6%	33.9%
A	NTLM	1338	Best64	37.0%	41.6%	43.0%	43.7%	44.2%
A	NTLM	1338	T0XICv1	58.4%	62.0%	63.5%	64.3%	64.8%
A	NTLM	1338	OneRule	65.4%	69.1%	70.0%	70.3%	70.6%
B	NTLM	1702	(none)	9.0%	12.6%	13.1%	13.3%	13.6%
B	NTLM	1702	Best64	17.6%	21.5%	22.2%	22.5%	22.7%
B	NTLM	1702	T0XICv1	38.9%	42.9%	43.7%	44.4%	44.9%
B	NTLM	1702	OneRule	50.2%	52.9%	53.7%	54.3%	54.8%

Table 18: Maximum percentage of retrieved hashes when using the best possible combination of one or more approaches before the introduction of NLWOVEN.