For password cracking, we used John the Ripper and managed to crack all of the passwords except for one of the UNIX Hashes (The 13 character alphanumeric one).

The resulting plain text passwords as shown in john.pot are:
1VvvHBSPu$n1ROeOtKnZB0XMpW4hTVw1:incongruous
NTb3df592fe797fce7e8823ce6ddeeed74:influenza
$1$4o7bNmjd$NmiWV/FK43BhFh01TVCUB1:ISSA%PLIYEV
1EoIFyokx$VF/7lYZ3K.4kbCzHozDjS/:acropho|3ia10
1vSFZFfyE$ljEGU3Qzej9q3HQcula2B1:ec7da897
LMfa5c528036ca1353:INFLUEN
LMf8ebbd4fa5799b41:ZA
LMdbfceb835d7c5385:WIE
LM2456a2bafcc64c37:BIV6
LM82290839e1916a4b:DAVIDBO
LMa8d0aea97dedbe89:6FV6XCP

The **UNIX** passwords found were:
"incongruous"
"ISSA%PLIYEV"
"acropho|3ia10"
"Ec7da897"

And the **Windows** passwords found were:
"influenza"
"DavidBowie"
"6FV6XCPBIV6"

The cracked passwords were weak due to a combination (to some degree) of the following reasons:

- They were not very long
- They used sensible known words/names
- They were made up of mostly alphabet characters and numbers

To be made stronger, the passwords would need to be longer, have more symbols, large numbers, and or non-sensible character combinations. This would force an attacker to have to try all possible combinations of characters (brute force) which can take incredible amounts of computation power to finally guess the right password.

We think that the passwords ISSA%PLIYEV, acropho|3ia10, and ec7da897 were relatively strong passwords and probably would not have been cracked without the hints. But given the hints, finding proper lists and specific methods to crack these wasn't difficult. The 13 character alphanumeric UNIX password was by far the most secure and we were unable to crack that even with the given hint.

The Windows passwords were much easier to crack. This is because of LANMAN's shortcomings of having a max of 14 chars, splitting that into 2 separate 7 character hashes, and by not distinguishing between character cases. Not having character cases instantly reduces the set of possible chars to try by 26 which makes the amount of possible combinations of 14 character 95^14. But since the password is split into 2 seven character hashes the difficulty of cracking is greatly reduced again to 2(95^7). Brute-forcing each 7 character half individually, even including symbols would probably take at the absolute worst a few days on the lab machine.
Information about the LANMAN has was obtained from wikipedia.

The Strategy used to crack each password is explained in detail below:

"1VvvHBSPu$n1ROeOtKnZB0XMpW4hTVw1:incongruous" and "NTb3df592fe797fce7e8823ce6ddeeed74:influenza"
These two were cracked by using lists of 11 and 9 character words that were pulled from the linux dictionary with the python script "grabwords.py".
The commands used were: "./john --wordlist=./in/eleven ./in/UNIX_hashes" and "./john --wordlist=./in/eleven ./in/Windows_Hashes --format=NT --fork=8"
These only took a few seconds each for john to crack.

$1$4o7bNmjd$NmiWV/FK43BhFh01TVCUB1:ISSA%PLIYEV
This one was cracked by using a list of roman and russian general names (generals.txt) which was created by copy pasting the different subdirectories for Roman and Russian generals found on wikipedia into a text file then using a python script (generalmix.py) to list the first and last names of each and mix the different cases possibilities of all lower, all upper, first upper for each name.
This also added in lastname firstname because initial attempts failed because soviet generals were missing. The special character was placed in between the names by using a custom wordlist rule that appears in john.conf as:
###
[List.Rules:General]
/?w s?w[%_$]
###
The command used was: "./john --wordlist=generals.txt --rules:General ./in/UNIX_hashes"

1EoIFyokx$VF/7lYZ3K.4kbCzHozDjS/:acropho|3ia10
This one was cracked by using a ten letter word list (ten.txt) created the same way as the nine and eleven ones. A John the Ripper custom wordlist ruleset was used to insert the 2 leet character replacements (found here). leet character replacements (found at:). There is a different rule for each leet character replacement. Each one will find the desired char

to replace (reject the word if it does not contain the char), delete the char at that position, insert the 2 character leet string into that position, and then append a 2 digit number to the end.

The rules look like this in the john.conf:

```
###
[List.Rules:Leet]
#a
%1a u Dp Ap"(L" $[0-9]$[0-9]
%1a u Dp Ap"/\" $[0-9]$[0-9]
#b
%1b u Dp Ap"|3" $[0-9]$[0-9]
%1b u Dp Ap"13" $[0-9]$[0-9]
%1b u Dp Ap"!3" $[0-9]$[0-9]
%1b u Dp Ap"(3" $[0-9]$[0-9]
%1b u Dp Ap"/3" $[0-9]$[0-9]
%1b u Dp Ap")3" $[0-9]$[0-9]
%1b u Dp Ap"j3" $[0-9]$[0-9]
#c
#%1c u Dp Ap"|=" $[0-9]$[0-9]
#d
%1d u Dp Ap"|)" $[0-9]$[0-9]
%1d u Dp Ap"(|" $[0-9]$[0-9]
%1d u Dp Ap"\[)" $[0-9]$[0-9]
%1d u Dp Ap"|>" $[0-9]$[0-9]
%1d u Dp Ap"T)" $[0-9]$[0-9]
%1d u Dp Ap"|7" $[0-9]$[0-9]
%1d u Dp Ap"c|" $[0-9]$[0-9]
%1d u Dp Ap"|}" $[0-9]$[0-9]
%1d u Dp Ap"|\\]" $[0-9]$[0-9]
#e
%1e u Dp Ap"\[-" $[0-9]$[0-9]
#f
%1f u Dp Ap"|=" $[0-9]$[0-9]
%1d u Dp Ap"ph" $[0-9]$[0-9]
%1d u Dp Ap"/=" $[0-9]$[0-9]
%1d u Dp Ap"|#" $[0-9]$[0-9]
#g
%1g u Dp Ap"C-" $[0-9]$[0-9]
%1g u Dp Ap"\[," $[0-9]$[0-9]
%1g u Dp Ap"{," $[0-9]$[0-9]
%1g u Dp Ap"<-" $[0-9]$[0-9]
%1g u Dp Ap"(." $[0-9]$[0-9]
```

```
#h
%1h u Dp Ap"}{" $[0-9]$[0-9]
#i
%1i u Dp Ap"\[\]" $[0-9]$[0-9]
%1i u Dp Ap"\]\[" $[0-9]$[0-9]
#j
%1j u Dp Ap"_|" $[0-9]$[0-9]
%1j u Dp Ap"_\\]" $[0-9]$[0-9]
#k
%1k u Dp Ap"|<" $[0-9]$[0-9]
%1k u Dp Ap"1<" $[0-9]$[0-9]
#l
%1l u Dp Ap"|_" $[0-9]$[0-9]
#m
%1m u Dp Ap"nn" $[0-9]$[0-9]
%1m u Dp Ap"^ ^" $[0-9]$[0-9]
#n
%1n u Dp Ap"^/" $[0-9]$[0-9]
%1n u Dp Ap"|V" $[0-9]$[0-9]
%1n u Dp Ap"/V" $[0-9]$[0-9]
#o
%1o u Dp Ap"()" $[0-9]$[0-9]
%1o u Dp Ap"\[\]" $[0-9]$[0-9]
%1o u Dp Ap"<>" $[0-9]$[0-9]
#p
%1p u Dp Ap"|*" $[0-9]$[0-9]
%1p u Dp Ap"|o" $[0-9]$[0-9]
%1p u Dp Ap"|^" $[0-9]$[0-9]
%1p u Dp Ap"|>" $[0-9]$[0-9]
%1p Dp Ap'|\"' $[0-9]$[0-9]
%1p u Dp Ap"|°" $[0-9]$[0-9]
%1p u Dp Ap"|º" $[0-9]$[0-9]
%1p u Dp Ap"|7" $[0-9]$[0-9]
#q
%1q u Dp Ap"0_" $[0-9]$[0-9]
%1q u Dp Ap"<|" $[0-9]$[0-9]
#r
%1r u Dp Ap"|~" $[0-9]$[0-9]
%1r u Dp Ap"|`" $[0-9]$[0-9]
%1r u Dp Ap"|~" $[0-9]$[0-9]
%1r u Dp Ap"|?" $[0-9]$[0-9]
%1r u Dp Ap"/2" $[0-9]$[0-9]
%1r u Dp Ap"|^" $[0-9]$[0-9]
```

```
%1r u Dp Ap"lz" $[0-9]$[0-9]
%1r u Dp Ap"|9" $[0-9]$[0-9]
%1r u Dp Ap"12" $[0-9]$[0-9]
%1r u Dp Ap"\[z" $[0-9]$[0-9]
%1r u Dp Ap".-" $[0-9]$[0-9]
%1r u Dp Ap"|2" $[0-9]$[0-9]
%1r u Dp Ap"|-" $[0-9]$[0-9]
#s
%1s u Dp Ap"es" $[0-9]$[0-9]
#t
%1t u Dp Ap"\[\]" $[0-9]$[0-9]
#u
%1u u Dp Ap"L|" $[0-9]$[0-9]
#v
%1v u Dp Ap"\/" $[0-9]$[0-9]
%1v u Dp Ap"|/" $[0-9]$[0-9]
%1v u Dp Ap"\\|" $[0-9]$[0-9]
#w
%1w u Dp Ap"VV" $[0-9]$[0-9]
%1w u Dp Ap"\\N" $[0-9]$[0-9]
%1w u Dp Ap"uu" $[0-9]$[0-9]
%1w u Dp Ap"2u" $[0-9]$[0-9]
%1w u Dp Ap"2v" $[0-9]$[0-9]
%1w u Dp Ap"v²" $[0-9]$[0-9]
#x
%1x u Dp Ap"><" $[0-9]$[0-9]
%1x u Dp Ap"}{" $[0-9]$[0-9]
%1x u Dp Ap")(" $[0-9]$[0-9]
%1x u Dp Ap"\]\[" $[0-9]$[0-9]
#y
%1y u Dp Ap"`/" $[0-9]$[0-9]
%1y u Dp Ap"'/" $[0-9]$[0-9]
%1y u Dp Ap"\V" $[0-9]$[0-9]
#z
%1z u Dp Ap"7_" $[0-9]$[0-9]
%1z u Dp Ap">_" $[0-9]$[0-9]

#UPPERCASE ----------

#A
%1A u Dp Ap"(L" $[0-9]$[0-9]
%1A u Dp Ap"/\" $[0-9]$[0-9]
#B
```

%1B u Dp Ap"|3" $[0-9]$[0-9]
%1B u Dp Ap"13" $[0-9]$[0-9]
%1B u Dp Ap"!3" $[0-9]$[0-9]
%1B u Dp Ap"(3" $[0-9]$[0-9]
%1B u Dp Ap"/3" $[0-9]$[0-9]
%1B u Dp Ap")3" $[0-9]$[0-9]
%1B u Dp Ap"J3" $[0-9]$[0-9]
#C
#%1C u Dp Ap"|=" $[0-9]$[0-9]
#D
%1D u Dp Ap"|)" $[0-9]$[0-9]
%1D u Dp Ap"(|" $[0-9]$[0-9]
%1D u Dp Ap"\[)" $[0-9]$[0-9]
%1D u Dp Ap"|>" $[0-9]$[0-9]
%1D u Dp Ap"T)" $[0-9]$[0-9]
%1D u Dp Ap"|7" $[0-9]$[0-9]
%1D u Dp Ap"C|" $[0-9]$[0-9]
%1D u Dp Ap"|}" $[0-9]$[0-9]
%1D u Dp Ap"|\\]" $[0-9]$[0-9]
#E
%1E u Dp Ap"\[-" $[0-9]$[0-9]
#F
%1F u Dp Ap"|=" $[0-9]$[0-9]
%1D u Dp Ap"pH" $[0-9]$[0-9]
%1D u Dp Ap"/=" $[0-9]$[0-9]
%1D u Dp Ap"|#" $[0-9]$[0-9]
#G
%1G u Dp Ap"C-" $[0-9]$[0-9]
%1G u Dp Ap"\[," $[0-9]$[0-9]
%1G u Dp Ap"{," $[0-9]$[0-9]
%1G u Dp Ap"<-" $[0-9]$[0-9]
%1G u Dp Ap"(." $[0-9]$[0-9]
#H
%1H u Dp Ap"}{" $[0-9]$[0-9]
#I
%1I u Dp Ap"\[\]" $[0-9]$[0-9]
%1I u Dp Ap"\]\[" $[0-9]$[0-9]
#J
%1J u Dp Ap"_|" $[0-9]$[0-9]
%1J u Dp Ap"_\\]" $[0-9]$[0-9]
#K
%1K u Dp Ap"|<" $[0-9]$[0-9]
%1K u Dp Ap"1<" $[0-9]$[0-9]

#L
%1L u Dp Ap"|_" $[0-9]$[0-9]
#M
%1M u Dp Ap"NN" $[0-9]$[0-9]
%1M u Dp Ap"^ ^" $[0-9]$[0-9]
#N
%1N u Dp Ap"^/" $[0-9]$[0-9]
%1N u Dp Ap"|V" $[0-9]$[0-9]
%1N u Dp Ap"/V" $[0-9]$[0-9]
#O
%1O u Dp Ap"()" $[0-9]$[0-9]
%1O u Dp Ap"\[\]" $[0-9]$[0-9]
%1O u Dp Ap"<>" $[0-9]$[0-9]
#P
%1P u Dp Ap"|*" $[0-9]$[0-9]
%1P u Dp Ap"|O" $[0-9]$[0-9]
%1P u Dp Ap"|^" $[0-9]$[0-9]
%1P u Dp Ap"|>" $[0-9]$[0-9]
%1P Dp Ap'|\'" $[0-9]$[0-9]
%1P u Dp Ap"|°" $[0-9]$[0-9]
%1P u Dp Ap"|º" $[0-9]$[0-9]
%1P u Dp Ap"|7" $[0-9]$[0-9]
#Q
%1Q u Dp Ap"0_" $[0-9]$[0-9]
%1Q u Dp Ap"<|" $[0-9]$[0-9]
#R
%1R u Dp Ap"|~" $[0-9]$[0-9]
%1R u Dp Ap"|`" $[0-9]$[0-9]
%1R u Dp Ap"|~" $[0-9]$[0-9]
%1R u Dp Ap"|?" $[0-9]$[0-9]
%1R u Dp Ap"/2" $[0-9]$[0-9]
%1R u Dp Ap"|^" $[0-9]$[0-9]
%1R u Dp Ap"LZ" $[0-9]$[0-9]
%1R u Dp Ap"|9" $[0-9]$[0-9]
%1R u Dp Ap"12" $[0-9]$[0-9]
%1R u Dp Ap"\[Z" $[0-9]$[0-9]
%1R u Dp Ap".-" $[0-9]$[0-9]
%1R u Dp Ap"|2" $[0-9]$[0-9]
%1R u Dp Ap"|-" $[0-9]$[0-9]
#S
%1S u Dp Ap"ES" $[0-9]$[0-9]
#T
%1T u Dp Ap"\[\]" $[0-9]$[0-9]

```
#U
%1U u Dp Ap"L|" $[0-9]$[0-9]
#V
%1V u Dp Ap"\/" $[0-9]$[0-9]
%1V u Dp Ap"|/" $[0-9]$[0-9]
%1V u Dp Ap"\\|" $[0-9]$[0-9]
#W
%1W u Dp Ap"VV" $[0-9]$[0-9]
%1W u Dp Ap"\\N" $[0-9]$[0-9]
%1W u Dp Ap"UU" $[0-9]$[0-9]
%1W u Dp Ap"2U" $[0-9]$[0-9]
%1W u Dp Ap"2V" $[0-9]$[0-9]
%1W u Dp Ap"V²" $[0-9]$[0-9]
#X
%1X u Dp Ap"><" $[0-9]$[0-9]
%1X u Dp Ap"}{" $[0-9]$[0-9]
%1X u Dp Ap")(" $[0-9]$[0-9]
%1X u Dp Ap"\]\[" $[0-9]$[0-9]
#Y
%1Y u Dp Ap"`/" $[0-9]$[0-9]
%1Y u Dp Ap"'/" $[0-9]$[0-9]
%1Y u Dp Ap"\V" $[0-9]$[0-9]
#Z
%1Z u Dp Ap"7_" $[0-9]$[0-9]
%1Z u Dp Ap">_" $[0-9]$[0-9]
###
```

The command used was: "./john --wordlist=ten.txt --rules:Leet ./in/UNIX_hashes"

1vSFZFfyE$ljEGU3Qzej9q3HQcula2B1:ec7da897

This one was cracked by using the pre-made "[List.External:DumbForce]" external rule that comes as part of the jumbo jtr package by modifying it to only use 0-9 and a-f for 8 characters. This was simply done by commenting out its character choices and inserting the following lines to only add the desired characters:

```
###
        i = 0;
        c = 0x30;
        while (c <= 0x39)
                charset[i++] = c++;
        c = 0x61;
        while (c <= 0x66)
                charset[i++] = c++;
###
```

The command used was: "./john --external:DumbForce ./in/UNIX_hashes"

LMfa5c528036ca1353:INFLUEN LMf8ebbd4fa5799b41:ZA
LMdbfceb835d7c5385:WIE LM82290839e1916a4b:DAVIDBO

The last 2 windows passwords were cracked by splitting the Windows_hashes into their separate LANMAN hashes. After reading the wikipedia article and already knowing that one of them was influenza by treating it as an NT hash, testing was done to try to find the "influen" and "za" pieces of the LM hash (test.txt).

This was found to work by simply splitting the hashes into 2 lines at the ":". After this List.External:Dumbforce was modified to start a 1 character and use the characters 0-9, A-Z.

Running: "./john --external:DumbForce ./in/Windows_Hashes --forman=LM --fork=8" very quickly found LMdbfceb835d7c5385:WIE and LM2456a2bafcc64c37:BIV6 which thanks to the hints were known to be the last 3 chars of the dead musician and the last 4 chars of the 11 char alphanumeric respectively.

LM82290839e1916a4b:DAVIDBO

This was guessed (test.txt) based on finding the "WIE" by typing "Bowie" in google because that was the first celebrity sounding name that came to mind that ended in "wie". Either way though this would have eventually been found during/after the brute force of finding the first 7 chars of the 11 char alphanumeric.

LMa8d0aea97dedbe89:6FV6XCP

This first half of the 11 character alphanumeric password was found by using the same modified External.DumbForce set to length of 7.

The command used was: "./john --external:DumbForce ./in/Windows_Hashes --forman=LM --fork=8"
