

# Managing Enterprise Cybersecurity

## MIS 4596

Unit#13

# Agenda

- Introduction
- Hacker's workflow
- Password vulnerabilities
- Password security techniques
- Password cracking
- Warning
- Lab 5: Password cracking
- Password entropy

# Password Cracking...

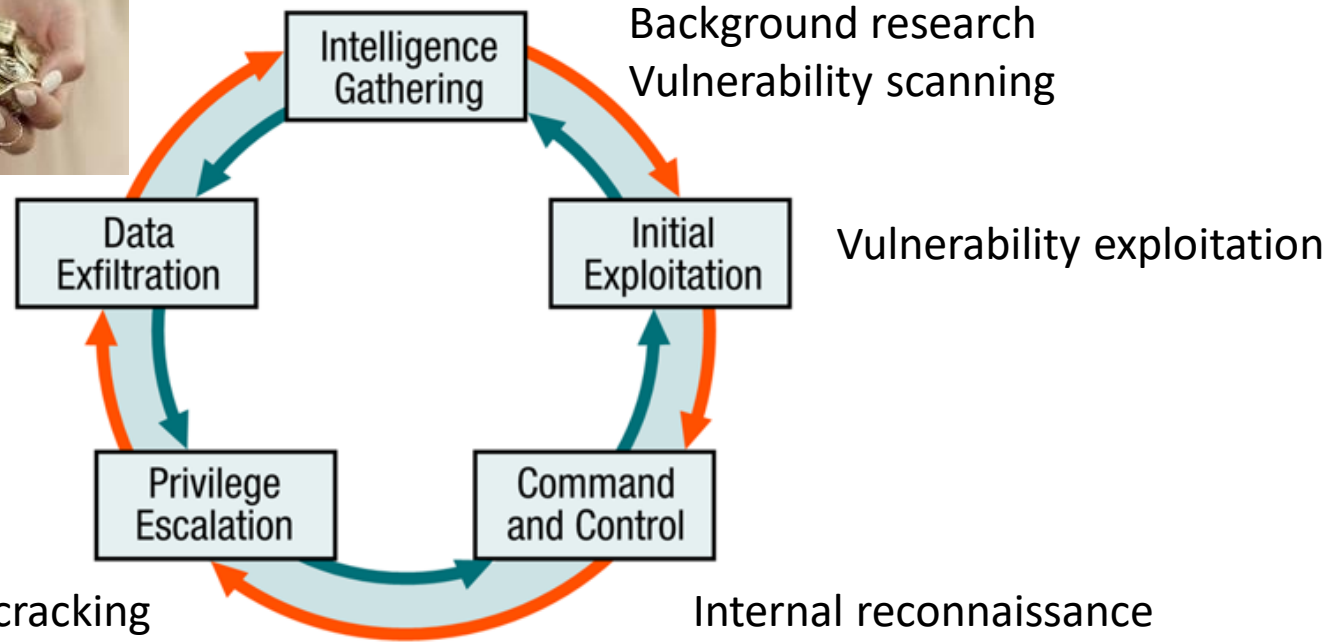
This week begins offensive security (e.g., ethical hacking & penetration testing)

- Explains techniques to crack (i.e., guess) passwords using a variety of techniques
- This lecture and the associated [Lab: Password Cracking](#) provides you with what you need to know to crack passwords on Milestone 3: Penetration Test

In Milestone 3, you may have the opportunity to crack passwords in:

- A cryptographic protocol for operating network services securely over an unsecured network
- A local Linux server's user accounts from the hashes found in /etc/shadow,
- Unsalted MD5 hashes in a local database
- And you will have the opportunity to decrypt an encrypted file by guessing its password
- ...

# Hacker's workflow – *Where Password Cracking fits in...*



A penetration test is an authorized simulated cyberattack on a computer system, performed to evaluate the security of the system

Colloquially known as a “pen test”, “pentest” or ethical hacking,

*Note: We are teaching Password Cracking now (before vulnerability scanning and exploitation), because knowing how to secure your passwords as soon as possible is essential to providing you personally relevant secure awareness training*

# Password

- A (hopefully) protected string of characters used to authenticate an individual
- Authentication factor based on what a person knows
- Is one of the most often used authentication mechanism



<b>#1</b>	Choose a phrase with at least 8 words. <i>This is my favorite sandwich in the world</i>
<b>#2</b>	Take the first letter of each word. <i>timfsitw</i>
<b>#3</b>	Switch one (or two) to an uppercase. <i>TimFsitw</i>
<b>#4</b>	Switch one to a number. <i>T1mFsitw</i>
<b>#5</b>	Switch one to a special character. <i>T1mF\$itw</i>
<b>#6</b>	Add something unique from each site. (i.e. add a b for banking, an f for facebook, etc.) <i>T1mF\$itwB</i>

# Password vulnerabilities

Although passwords are the most common authentication mechanism, they are one of the weakest security mechanisms available

Why?

- Users choose words easily guessed
- Users tell others their passwords
- Users write passwords down on a sticky note and hide it under the keyboard
- Many users consider security an unimportant uninteresting part of using computers – until someone steals their confidential information...
- This is why password policies are needed
  - With good policies passwords can provide effective security, if they are:
    - Properly generated, updated, and kept secret



# Techniques used to attack a password

- Electronic monitoring – Replay Attack
  - Listening to network traffic to capture data when a user is sending her password to an authentication server
  - The password can be copied and reused by the attacker at another time
- Access the password file
  - Usually done on an authentication server
  - Password file contains many users' passwords – if compromised can be a source of much damage
  - File should be protected with encryption and access control
- Brute-force attack
  - Performed with tools that cycle through many possible character, number, and symbol combinations to uncover a password
- Dictionary attack
  - Files of thousands of words are compared to the user's password until a match is found
- Rainbow table
  - An attacker uses a table that contains all possible passwords already in a hash format
- Social engineering
  - An attacker falsely convinces an individual that she has the necessary authorization to access specific resources

# Password Policy

Is a set of rules designed to enhance computer security by encouraging users to employ strong passwords and use them properly

- Is often part of an organization's official regulations and may be taught as part of security awareness training
- Either the password policy is merely advisory, or the computer systems force users to comply with it



## 11.15 - Password Policy and Guidelines

### 2. Responsibilities of Systems Processing Passwords

All WCM systems—including servers, applications, and websites that are hosted by or for WCM—must be designed to accept passwords and transmit them with proper safeguards.

- Passwords must be prohibited from being displayed when entered.
- Passwords must never be stored in clear, readable format (encryption must always be used).
- Passwords must never be stored as part of a login script, program, or automated process.
- Systems storing or providing access to confidential data or remote access to the internal network should be secured with multifactor authentication.
- Encrypted password hashes must never be accessible to unauthorized individuals.
- Where possible, salted hashes should be used for password encryption. Exceptions should be filed and reviewed on a regular basis.
- Where any of the above items are not supported, appropriate authorizations and access control methods must be implemented to ensure only a limited number of authorized individuals have access to readable passwords.

### 3. Password Requirements

The following parameters indicate the minimum requirements for passwords for all individual accounts where passwords are:

- At least eight (8) characters;
- Not based on anything somebody else could easily guess or obtain using person related information (e.g., names, CWID, telephone numbers, dates of birth, etc.);
- Not vulnerable to a dictionary attack (see Recommendations for Creating Compliant Passwords section); and,
- Effective July 5, 2017, a combination of at least one character from each of the following four listed character types (older passwords require at least one character from three of the following four types):
  - English uppercase letters (A-Z),
  - English lowercase letters (a-z)
  - Base 10 digits (0-9)
  - Non-alphanumeric (such as ` ~ ! @ # \$ % ^ & \* ( ) \_ + = { } | \ : " ; ' < > ? , . / and space)

### 4. Password Expiration

In order to prevent an attacker from making use of a password that may have been discovered, passwords are deemed temporary and must be changed regularly. ITS Security reserves the right to reset a user's password in the event a compromise is suspected or reported. The required frequency at which passwords must be changed varies based on the





# Additional Password Security Techniques

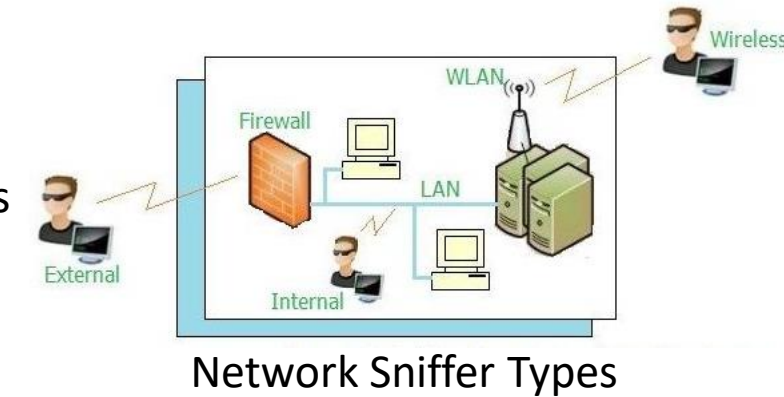
A systems' administrator can:

- Set operating parameters to test the strength of a password and only accept new passwords that meet password strength rules
- Set operating parameters locking the user out after a certain number of failed logon attempts
  - Users can be locked out for 5 minutes or a full day based on a threshold
- Turn on alerts sent to the user bringing attention to logon attempts using his/her credentials
  - After each successful logon, presenting the date and time of the last successful logon and if there were any unsuccessful logon attempts
- Turn on an audit trail to track password usage, and both successful and unsuccessful logon attempts
  - This audit information should include: date, time, user ID, and workstation user logged in from
- Force the user to change a password on a more frequent basis to assure the password will not be guessed by an intruder, remembering the password history and blocking reuse of passwords (e.g. last 5 or 10)
  - A balance between protection and practicality must be enforced – too often may lead to users forgetting their passwords and unnecessary management overhead

# Additional Password Security Techniques...

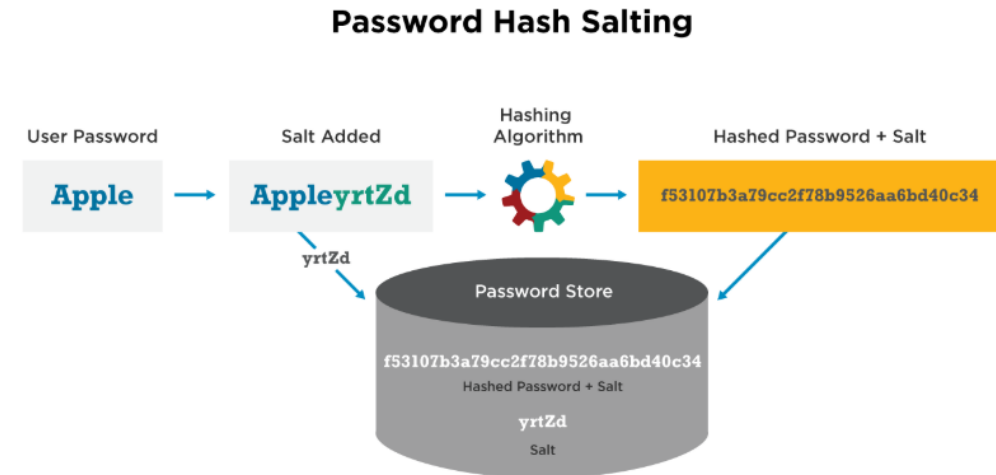
## Password hashing

- An attacker capturing (“sniffs”) a password from the network typically has more work to do before knowing the password value because many systems hash the password with a hashing algorithm to ensure passwords are not sent nor stored in cleartext
- Linux and Unix systems do not store plaintext passwords but run the passwords through a hash algorithm and place the hashed passwords in a “shadow” file system



## Password hashing with “salt”

- Unix systems can be set to add “salt” to its password hashing
- A salt is a random value concatenated with the password before the hashing to add more randomness and complexity into the encryption process
- This makes it harder for the hacker to decrypt and uncover the password
- Adding different salts into the hashing of the same password can result in several thousand different hashes – making it much more difficult for a hacker to attack the passwords



# Hash without salt

```
phillipnontenure@kali:~$ echo -n "monkey" | shasum  
ab87d24bdc7452e55738deb5f868e1f16dea5ace -
```



ab87d24bdc7452e55738deb5f868e1f16dea5ace



All Maps Videos Images Shopping More Settings Tools

About 410 results (0.43 seconds)

<https://hashtoolkit.com> › Decrypt Hash

**Best SHA1 Hash Password Decrypt | Hash Toolkit ...**

No hashes found for **ab87d24bdc7452e55738deb5f868e1f16dea5ace**. What is Hash Toolkit?  
Decrypt password hashes super fast with Hash Toolkit. You can ...

<https://sha1.gromweb.com> › hash=ab87d24bdc7452e55... :

**SHA-1 reverse for ...**

**ab87d24bdc7452e55738deb5f868e1f16dea5ace** was successfully reversed into the string:  
monkey. Feel free to provide some other SHA-1 hashes you would ...

<https://md5hashing.net> › hash › ab87d24bdc7452e557... :

**Hash sha1: ab87d24bdc7452e55738deb5f868e1f16dea5ace**

Nov 2, 2015 — Sha1: **ab87d24bdc7452e55738deb5f868e1f16dea5ace** hash digest (reversed,  
unhashed, decoded, decrypted). Md2 ...

<https://md5calc.com> › hash › sha1 › monkey :

**SHA1 hash for "monkey" is ... - Md5Calc.com**

SHA1 hash for "monkey" is "**ab87d24bdc7452e55738deb5f868e1f16dea5ace**". Free online sha1  
hash calculator. Calculate sha1 hash from string.

<https://www.youtube.com> › watch · Translate this page :

**ab87d24bdc7452e55738deb5f868e1f16dea5ace - YouTube**

Feb 14, 2020 — <https://hashhub.cat/sha1/ab87d24bdc74...>

**ab87d24bdc7452e55738deb5f868e1f16dea5ace** monkey. Category. Education. Show more

# Hash with a salt

```
phillipnontenure@kali:~$ echo -n "2u9fh928""monkey" | shasum  
95967387233515d497312ab1dc8f0d58f166198d -
```



95967387233515d497312ab1dc8f0d58f166198d



All

Maps

Videos

Images

Shopping

More

Settings

Tools

Your search - **95967387233515d497312ab1dc8f0d58f166198d** - did not match any documents.

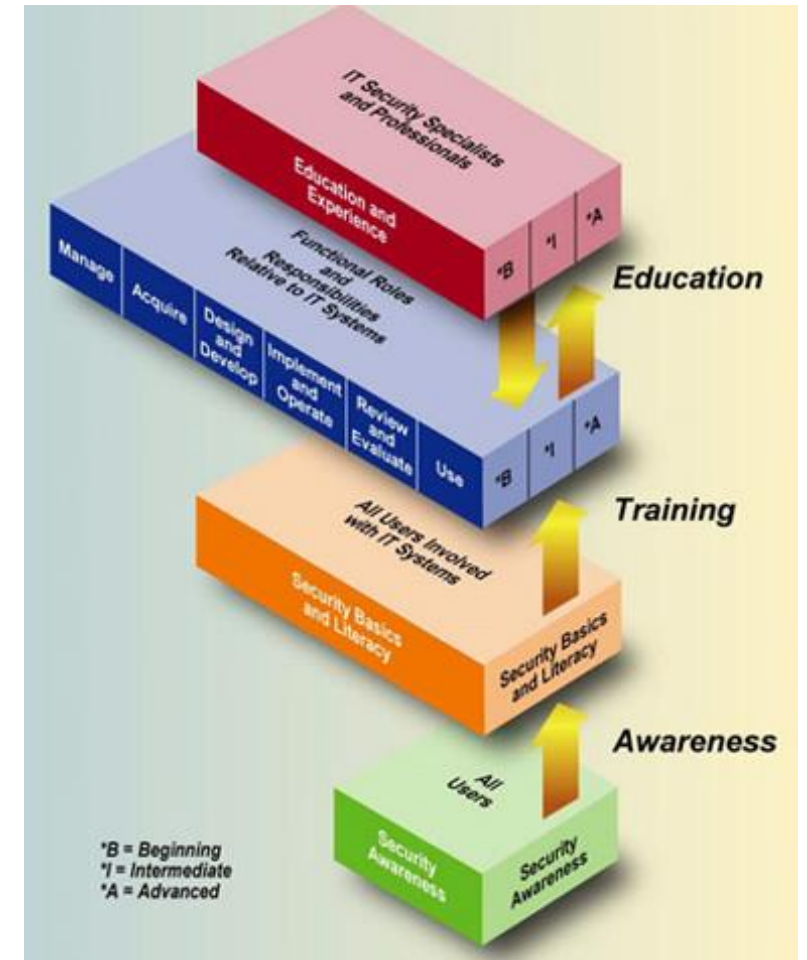
Suggestions:

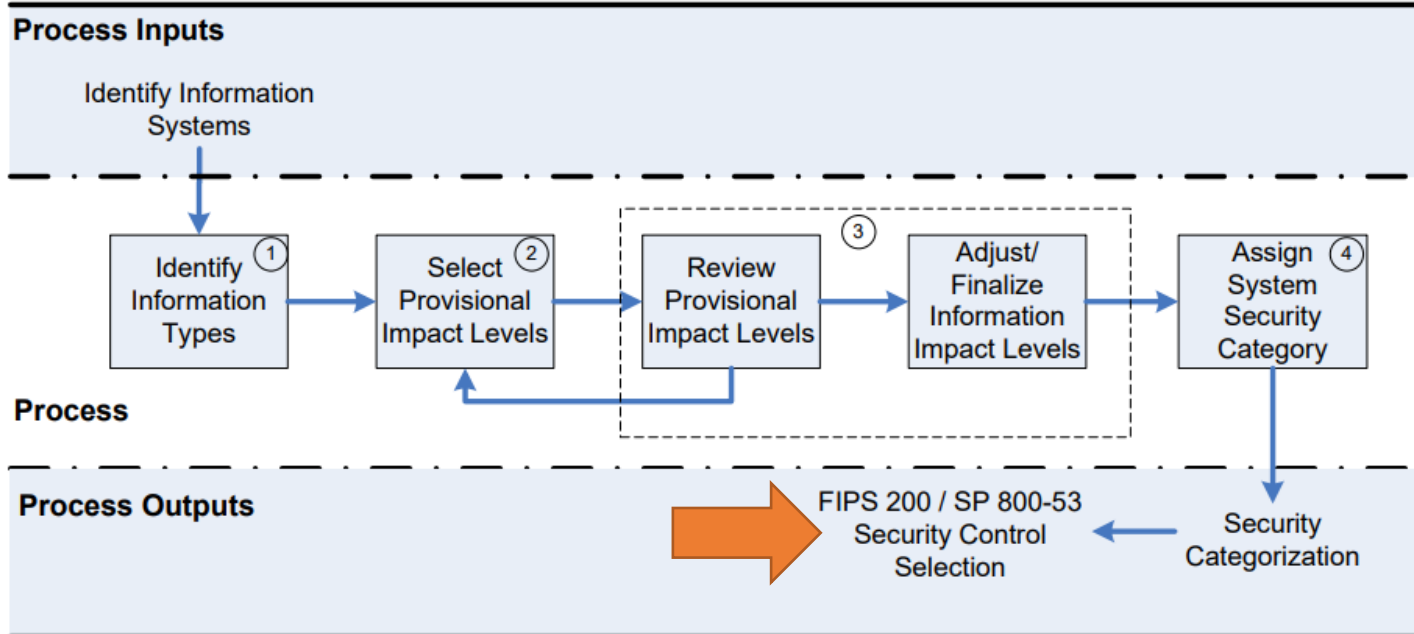
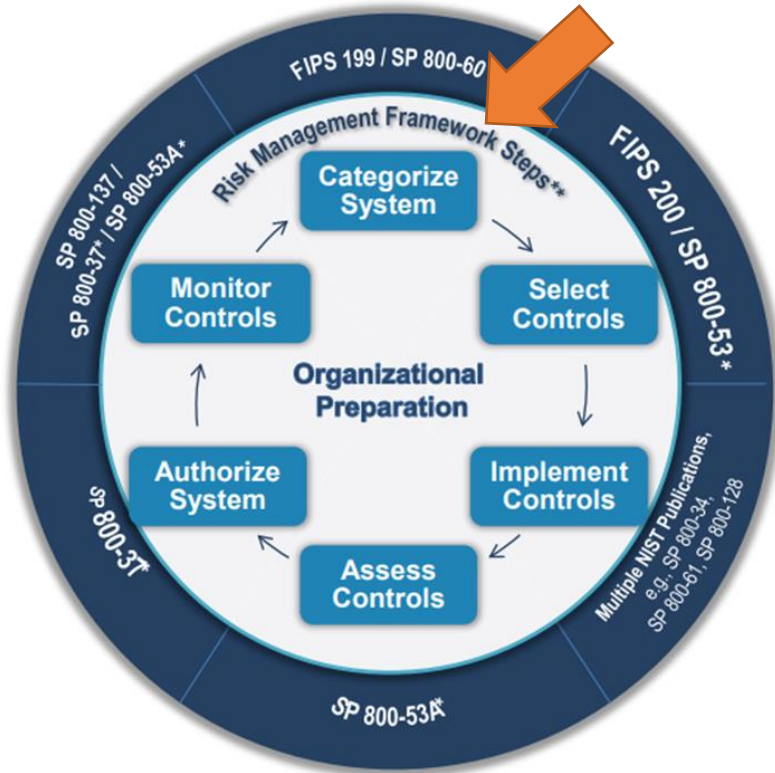
- Make sure all words are spelled correctly.
- Try different keywords.
- Try more general keywords.

# Additional Password Security Techniques...

## Education is the key!

- Users should be an extension of a security team, not the opposition
- Security awareness programs should help users
  - Understand what is expected of them
  - Why they should protect their passwords
  - How passwords can be stolen
  - How to protect passwords





NIST Special Publication 800-60 Volume I  
Revision 1

**NIST**  
National Institute of Standards and Technology  
U.S. Department of Commerce

**Volume I:  
Guide for Mapping Types of  
Information and Information  
Systems to Security Categories**

Kevin Stine  
Rich Kissel  
William C. Barker  
Jim Fahlsing  
Jessica Gulick

**INFORMATION SECURITY**

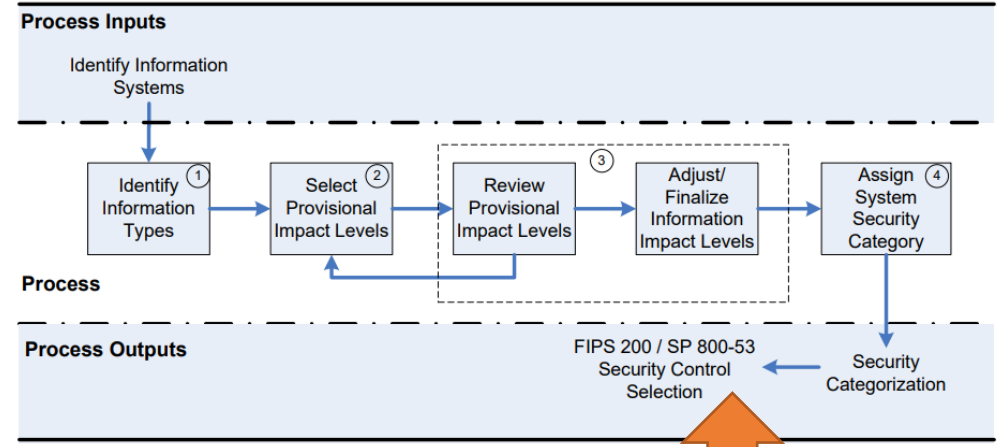
Computer Security Division  
Information Technology Laboratory  
National Institute of Standards and Technology  
Gaithersburg, MD 20899-8930

August 2008

**U.S. DEPARTMENT OF COMMERCE**  
*Carlos M. Gutierrez, Secretary*

**NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY**  
*James M. Turner, Deputy Director*

# NIST Special Publication 800-53



CLASS	FAMILY
Management	Risk Assessment
Management	Planning
Management	System and Services Acquisition
Management	Certification, Accreditation, and Security Assessments
Operational	Personnel Security
Operational	Physical and Environmental Protection
Operational	Contingency Planning
Operational	Configuration Management
Operational	Maintenance
Operational	System and Information Integrity
Operational	Media Protection
Operational	Incident Response
Operational	Awareness and Training
Technical	Identification and Authentication
Technical	Access Control
Technical	Audit and Accountability
Technical	System and Communications Protection




NIST Special Publication 800-53  
Revision 5

## Security and Privacy Controls for Information Systems and Organizations

JOINT TASK FORCE

This publication is available free of charge from:  
<https://doi.org/10.6028/NIST.SP.800-53r5>

September 2020  
INCLUDES UPDATES AS OF 12-10-2020; SEE PAGE XVII



U.S. Department of Commerce  
*Wilbur L. Ross, Jr., Secretary*

National Institute of Standards and Technology  
*Walter Copan, NIST Director and Under Secretary of Commerce for Standards and Technology*

# NIST Special Publication 800-53

## IA-5 AUTHENTICATOR MANAGEMENT

Identification and Authentication					
IA-1	Identification and Authentication Policy and Procedures	P1	IA-1	IA-1	IA-1
IA-2	Identification and Authentication (Organizational Users)	P1	IA-2 (1) (12)	IA-2 (1) (2) (3) (8) (11) (12)	IA-2 (1) (2) (3) (4) (8) (9) (11) (12)
IA-3	Device Identification and Authentication	P1	Not Selected	IA-3	IA-3
IA-4	Identifier Management	P1	IA-4	IA-4	IA-4
IA-5	Authenticator Management	P1	IA-5 (1) (11)	IA-5 (1) (2) (3) (11)	IA-5 (1) (2) (3) (11)
IA-6	Authenticator Feedback	P2	IA-6	IA-6	IA-6
IA-7	Cryptographic Module Authentication	P1	IA-7	IA-7	IA-7
IA-8	Identification and Authentication (Non-Organizational Users)	P1	IA-8 (1) (2) (3) (4)	IA-8 (1) (2) (3) (4)	IA-8 (1) (2) (3) (4)

### Control Enhancements:

#### (1) AUTHENTICATOR MANAGEMENT | [PASSWORD-BASED AUTHENTICATION](#)

##### For password-based authentication:

- Maintain a list of commonly-used, expected, or compromised passwords and update the list [*Assignment: organization-defined frequency*] and when organizational passwords are suspected to have been compromised directly or indirectly;
- Verify, when users create or update passwords, that the passwords are not found on the list of commonly-used, expected, or compromised passwords in IA-5(1)(a);
- Transmit passwords only over cryptographically-protected channels;
- Store passwords using an approved salted key derivation function, preferably using a keyed hash;
- Require immediate selection of a new password upon account recovery;
- Allow user selection of long passwords and passphrases, including spaces and all printable characters;
- Employ automated tools to assist the user in selecting strong password authenticators; and
- Enforce the following composition and complexity rules: [*Assignment: organization-defined composition and complexity rules*].

### Control: Manage system authenticators by:

- Verifying, as part of the initial authenticator distribution, the identity of the individual, group, role, service, or device receiving the authenticator;
- Establishing initial authenticator content for any authenticators issued by the organization;
- Ensuring that authenticators have sufficient strength of mechanism for their intended use;
- Establishing and implementing administrative procedures for initial authenticator distribution, for lost or compromised or damaged authenticators, and for revoking authenticators;
- Changing default authenticators prior to first use;
- Changing or refreshing authenticators [*Assignment: organization-defined time period by authenticator type*] or when [*Assignment: organization-defined events*] occur;
- Protecting authenticator content from unauthorized disclosure and modification;
- Requiring individuals to take, and having devices implement, specific controls to protect authenticators; and
- Changing authenticators for group or role accounts when membership to those accounts changes.



# Additional Password Security Techniques

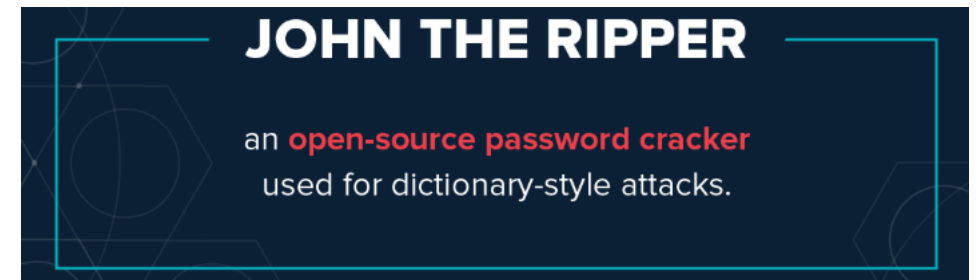
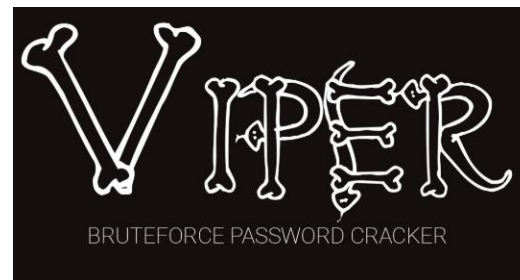
## Password Checker

- Is a tool used by a security professional to test the strength of a password
- Performs brute-force and/or dictionary attacks to detect weak passwords and fix them before an attacker find the vulnerabilities

## Password Cracker

- Is the same tool (i.e. Password Checker) but used by a hacker to uncover vulnerabilities to exploit before the security professional can fix them

Most security tools have this dual nature!



# Brutalis

“Fastest, meanest, most hardcore password cracker money can buy, ...clawing through hashes at unprecedented speeds

- \$21,169
- Ships with 3-year warranty and full commercial support
- Uses Graphics Processing Units (GPU)
  - Their highly parallel structure makes them more efficient than general-purpose central processing units (CPUs) for algorithms that process large blocks of data in parallel
- Uses
  - 8 NVIDIA GTX GPUs
  - 2 Intel Xeon E5-2600 v4 CPUs
  - Up to 3 TB memory
  - 18 TB Solid State Device storage



Terahash's [LOphtCrack](#)



# WARNING

Never use the tools you learn in this class to test, attack, or break a computer that you do own or have written permission to do so!

For example, you need to obtain management's approval before attempting to test employees' passwords with the intent of identifying weak passwords.

Explaining you are trying to help the situation, and not hurt it, after the fact is not a good situation to be in.

# Lab 7: Online & Offline password attacks

## Lab: Password Cracking

Part 1: Test Password Security

Part 2: Check an Account for a Prior Data Breach

Part 3. Sign-up for Two Factor Authentication

Part 4. Install and Set up a Password Manager

Part 5: Online Password Attack

Part 6: Offline Attack Using Hashcat

Part 7. Cracking LinkedIn Hashes Using Hashcat

Part 8. Secure Password Hashing

Part 9. Create a Targeted Wordlist Using CeWL

```
Terminal - root@kali: /home/geocryp4596
File Edit View Terminal Tabs Help
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "carla" - 1218 of 14344399 [child 2] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "australia" - 1219 of 14344399 [child 1] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "bismillah" - 1220 of 14344399 [child 4] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "7654321" - 1221 of 14344399 [child 10] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "bigdaddy" - 1222 of 14344399 [child 8] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "9876543210" - 1223 of 14344399 [child 12] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "photos" - 1224 of 14344399 [child 9] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "franklin" - 1225 of 14344399 [child 7] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "pink123" - 1226 of 14344399 [child 6] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "erick" - 1227 of 14344399 [child 11] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "vanilla" - 1228 of 14344399 [child 10] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "briana" - 1229 of 14344399 [child 13] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "hello123" - 1230 of 14344399 [child 14] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "jacob" - 1231 of 14344399 [child 1] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "hilary" - 1232 of 14344399 [child 3] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "pedro" - 1233 of 14344399 [child 5] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "loveme2" - 1234 of 14344399 [child 15] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "callum" - 1235 of 14344399 [child 11] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "watermelon" - 1236 of 14344399 [child 6] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "lourdes" - 1237 of 14344399 [child 1] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "janelle" - 1238 of 14344399 [child 4] (0/0)
[443][http-get] host: is.theorizeit.org login: istheory password: 9876543210
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-02-12 17:51:07
root@kali: /home/geocryp4596#
```



# Use Caution!

- Never test a password you are currently using and dependent on!
- If you wish to test your password's strength, try a “similarly structured” password
  - Similar length
  - Similar structured use of letters, capitals, special characters, numbers...

## Lab: Password Cracking

By Drs. Anthony Vance and Dave Eargle

This lab uses the following VMs:

- Kali

### Part 1: Test Password Security

1. Visit the following URL:

<https://lowe.github.io/tryxcvbn/>

This website estimates the strength of passwords you enter. Passwords never leave your browser – if you are so inclined, you can confirm as much by perusing the [source code](#) and/or by opening your browser developer tools and observing the (java)script that runs when you enter passwords.

2. Try out different passwords to see how strong they are.

Optional: If you want to learn more about password strength estimation, see [this video and paper](#).

### Part 2: Check an Account for a Prior Data Breach

1. Check to see if one of your online accounts has already been breached.

Visit: <https://haveibeenpwned.com>. Type in one of your email accounts or usernames to see if it has already been compromised in a data breach.

2. Next visit: <https://haveibeenpwned.com/Passwords>

Try out some passwords to see if they have already been compromised in a data breach.

3. Finally, visit: <https://haveibeenpwned.com/NotifyMe>

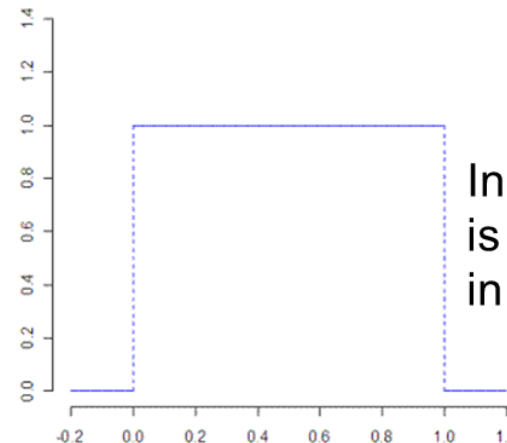
Sign up to be notified when one of your accounts is breached in the future.

Question: Was one of your accounts breached? If so, which one(s)?

# Password Entropy

Password entropy is a measurement of how unpredictable a password is

- Password strength is usually expressed in terms of "information entropy"
- A password based on a random selection of 64 bits (8 8-bit ASCII characters) would require  $2^{64}$  (18,446,744,073,709,551,616) attempts to exhaust all possibilities during a brute force search
  - Thus, by increasing the entropy of the password by one more randomly selected bit doubles the number of guesses, making an attacker's task twice as difficult
  - On average, an attacker will have to try half the possible number of passwords before finding the correct one



In a uniform distribution any number is equally likely, the average is right in the middle, or half the distribution

# Diceware Exercise – Part 1

In teams of 2:

- Role the 5 dice to generate a random 5-digit number

- <https://www.random.org/dice/>

You rolled 5 dice:



Timestamp: 2021-10-05 18:57:19 UTC

Roll Again

Go Back

- Use the 5-digit to select a “random” word from:

- [https://www.eff.org/files/2016/07/18/eff\\_large\\_wordlist.txt](https://www.eff.org/files/2016/07/18/eff_large_wordlist.txt)

11111	abacus
11112	abdomen
11113	abdominal
11114	abide
11115	abiding
11116	ability
11121	ablaze
11122	able
11123	abnormal
11124	abrasion
11125	abrasive
11126	abreast
11131	abridge
11132	abroad
11133	abruptly
11134	absence
11135	absentee
11136	absently
11141	absinthe
11142	absolute
11143	absolve
11144	abstain
11145	abstract
66634	zealous
66635	zebra
66636	zen
66641	zeppelin
66642	zero
66643	zestfully
66644	zesty
66645	zigzagged
66646	zipfile
66651	zipping
66652	zippy
66653	zips
66654	zit
66655	zodiac
66656	zombie
66661	zone
66662	zoning
66663	zookeeper
66664	zoologist
66665	zoology
66666	zoom



# Diceware Exercise



$$6^5 = 7776 = 2^{12.9}$$

The strength of a Diceware passphrase depends on how many words it contains. If you choose one word (out of a list of 7,776 words), an attacker has a 1 in 7,776 chance of guessing your word on the first try. To guess your word it will take an attacker at least one try, at most 7,776 tries, and on average 3,888 tries (because there's a 50 percent chance that an attacker will guess your word by the time they are halfway through the word list).

This means that with two words, there are  $7,776^2$ , or 60,466,176 different potential passphrases. On average, a two-word Diceware passphrase could be guessed after the first 30 million tries.

And a five-word passphrase, which would have  $7,776^5$  possible passphrases, could be guessed after an average of 14 quintillion tries (a 14 with 18 zeroes).

The amount of uncertainty in a passphrase (or in an encryption key, or in any other type of information) is measured in bits of entropy. You can measure how secure your random passphrase is by how many bits of entropy it contains.

Each word from the Diceware list is worth about 12.92 bits of entropy (because  $2^{12.92}$  is about 7,776). Therefore, if you choose seven words – you will end up with a passphrase with about 90.5 bits of entropy (because 12.92 times seven is about 90.5).

<https://theintercept.com/2015/03/26/passphrases-can-memorize-attackers-cant-guess/>

# Diceware Exercise – Part 2

- Answer the following questions:

1. Using dice, create two five-word Diceware passphrases. What passphrases did you create?
  2. What is the amount of entropy in bits for each of your five-word Diceware passphrases? How many possible passwords does that represent?
1. On average, how long in hours would it take to guess a five-word Diceware passphrase if you could try 1 billion passwords a second? How many years? Show your work.

*Note: Each Diceware word is worth an additional 12.9 bits of entropy. This is because  $2^{12.9} \approx 7776$ , the length of the Diceware word list.*

*Note: The average of a uniform distribution is half the numbers in a set.*

# Agenda

- ✓ Introduction
- ✓ Hacker's workflow
- ✓ Password vulnerabilities
- ✓ Password security techniques
- ✓ Password cracking
- ✓ Warning
- ✓ Lab 5: Password cracking
  - ✓ Caution
- ✓ Password entropy