Linux Network Systems Administration - Commands and Expected Outputs

# 1. Aspect - Description

| **Command**  Extra Aspect Description (Meas or Judg) OR Judgement Score Description (Judg only) |
| --- |
| **Expected Output**  Requirement (Measurement Only) |

# 2. LDAP: OU Employees exists

| **Command**  ldapsearch -H ldap://localhost -b dc=int,dc=worldskills,dc=org -x "(&(objectclass=organizationalunit)(ou=Employees))" -D cn=admin,dc=int,dc=worldskills,dc=org -w Skill39@Lyon |
| --- |
| **Expected Output**  > # extended LDIF # # LDAPv3 # base <dc=int,dc=worldskills,dc=org> with scope subtree # filter: (&(objectclass=organizationalunit)(ou=Employees)) # requesting: ALL #  # Employees, int.worldskills.org dn: ou=Employees,dc=int,dc=worldskills,dc=org ou: Employees objectClass: organizationalUnit  # search result search: 2 result: 0 Success  # numResponses: 2 # numEntries: 1 |

# 3. LDAP: User jamie, peter & admin exists

| **Command**  $ (ldapsearch -H ldap://localhost -b dc=int,dc=worldskills,dc=org -x "(&(objectclass=person)(uid=jamie))" -D cn=admin,dc=int,dc=worldskills,dc=org -w Skill39@Lyon || ldapsearch -H ldap://localhost -b dc=int,dc=worldskills,dc=org -x "(&(objectclass=person)(uid=jamie))") 2>&1 $ (ldapsearch -H ldap://localhost -b dc=int,dc=worldskills,dc=org -x "(&(objectclass=person)(uid=peter))" -D cn=admin,dc=int,dc=worldskills,dc=org -w Skill39@Lyon || ldapsearch -H ldap://localhost -b dc=int,dc=worldskills,dc=org -x "(&(objectclass=person)(uid=peter))") 2>&1 $ (ldapsearch -H ldap://localhost -b cn=admin,dc=int,dc=worldskills,dc=org -x "(objectclass=\*)" -D cn=admin,dc=int,dc=worldskills,dc=org -w Skill39@Lyon || ldapsearch -H ldap://localhost -b cn=admin,dc=int,dc=worldskills,dc=org -x "(objectclass=\*)") 2>&1 |
| --- |
| **Expected Output**  # jamie, Employees, int.worldskills.org dn: cn=jamie,ou=Employees,dc=int,dc=worldskills,dc=org cn: jamie sn: Oliver givenName: Jamie mail: jamie.oliver@dmz.worldskills.org uid: jamie uidNumber: 1111 gidNumber: 1111 userPassword:: e1NTSEF9RzgxeGxDblZqZVh1NEp6bVcxSGhXc1NCRFg2UENjTU8= homeDirectory: /home/jamie objectClass: inetOrgPerson objectClass: posixAccount objectClass: top  # search result search: 2 result: 0 Success  # numResponses: 2 # numEntries: 1   # peter, Employees, int.worldskills.org dn: cn=peter,ou=Employees,dc=int,dc=worldskills,dc=org cn: peter sn: Fox givenName: Peter mail: peter.fox@dmz.worldskills.org uid: peter uidNumber: 1113 gidNumber: 1113 userPassword:: e1NTSEF9RzgxeGxDblZqZVh1NEp6bVcxSGhXc1NCRFg2UENjTU8= homeDirectory: /dev/null/nohome objectClass: inetOrgPerson objectClass: posixAccount objectClass: top  # search result search: 2 result: 0 Success  # numResponses: 2 # numEntries: 1  # admin, int.worldskills.org dn: cn=admin,dc=int,dc=worldskills,dc=org sn: Admin cn: Admin objectClass: top objectClass: person  # search result search: 2 result: 0 Success  # numResponses: 2 # numEntries: 1 |

# 4. LDAP: Users jamie and peter have uid, mail attribute and correct OU

| **Command**  $ (ldapsearch -H ldap://localhost -b dc=int,dc=worldskills,dc=org -x "(&(objectclass=person)(uid=jamie))" -D cn=admin,dc=int,dc=worldskills,dc=org -w Skill39@Lyon || ldapsearch -H ldap://localhost -b dc=int,dc=worldskills,dc=org -x "(&(objectclass=person)(uid=jamie))") 2>&1 $ (ldapsearch -H ldap://localhost -b dc=int,dc=worldskills,dc=org -x "(&(objectclass=person)(uid=peter))" -D cn=admin,dc=int,dc=worldskills,dc=org -w Skill39@Lyon || ldapsearch -H ldap://localhost -b dc=int,dc=worldskills,dc=org -x "(&(objectclass=person)(uid=peter))") 2>&1 |
| --- |
| **Expected Output**  # jamie, Employees, int.worldskills.org dn: cn=jamie,ou=Employees,dc=int,dc=worldskills,dc=org cn: jamie sn: Oliver givenName: Jamie mail: jamie.oliver@dmz.worldskills.org uid: jamie uidNumber: 1111 gidNumber: 1111 userPassword:: e1NTSEF9RzgxeGxDblZqZVh1NEp6bVcxSGhXc1NCRFg2UENjTU8= homeDirectory: /home/jamie objectClass: inetOrgPerson objectClass: posixAccount objectClass: top  # search result search: 2 result: 0 Success  # numResponses: 2 # numEntries: 1   # peter, Employees, int.worldskills.org dn: cn=peter,ou=Employees,dc=int,dc=worldskills,dc=org cn: peter sn: Fox givenName: Peter mail: peter.fox@dmz.worldskills.org uid: peter uidNumber: 1113 gidNumber: 1113 userPassword:: e1NTSEF9RzgxeGxDblZqZVh1NEp6bVcxSGhXc1NCRFg2UENjTU8= homeDirectory: /dev/null/nohome objectClass: inetOrgPerson objectClass: posixAccount objectClass: top  # search result search: 2 result: 0 Success  # numResponses: 2 # numEntries: 1 |

# 5. LDAP: Users jamie, peter and admin can login on LDAP

| **Command**  $ ldapsearch -H ldap://localhost -b dc=int,dc=worldskills,dc=org -x "(&(objectclass=person)(uid=jamie))" -D cn=jamie,ou=Employees,dc=int,dc=worldskills,dc=org -w Skill39@Lyon $ ldapsearch -H ldap://localhost -b dc=int,dc=worldskills,dc=org -x "(&(objectclass=person)(uid=peter))" -D cn=peter,ou=Employees,dc=int,dc=worldskills,dc=org -w Skill39@Lyon $ ldapsearch -H ldap://localhost -b cn=admin,dc=int,dc=worldskills,dc=org -x "(objectclass=\*)" -D cn=admin,dc=int,dc=worldskills,dc=org -w Skill39@Lyon |
| --- |
| **Expected Output**  3x Exit code was 0 |

# 6. CA: Root CA exists and has correct attributes

| **Command**  $ openssl x509 -in /opt/grading/ca/ca.pem -noout -subject -ext basicConstraints,keyUsage |
| --- |
| **Expected Output**  subject=CN = ClearSky Root CA X509v3 Key Usage: critical  Certificate Sign X509v3 Basic Constraints: critical  CA:TRUE |

# 7. CA: Services CA exists and has correct attributes

| **Command**  $ openssl x509 -in /opt/grading/ca/ca.pem -noout -ext basicConstraints,keyUsage $ openssl verify -CAfile /opt/grading/ca/ca.pem /opt/grading/ca/services.pem |
| --- |
| **Expected Output**  > X509v3 Key Usage: critical  Certificate Sign X509v3 Basic Constraints: critical  CA:TRUE  > /opt/grading/ca/services.pem: OK |

# 8. CA: Webserver Certificate exists and has correct attributes

| **Command**  $ openssl x509 -in /opt/grading/ca/web.pem -noout -subject $ openssl verify -CAfile <(cat /opt/grading/ca/services.pem /opt/grading/ca/ca.pem) /opt/grading/ca/web.pem |
| --- |
| **Expected Output**  > subject=CN = www.dmz.worldskills.org  > /opt/grading/ca/web.pem: OK |

# 9. CA: Mailserver Certificate exists and has the correct attributes

| **Command**  $ openssl x509 -in /opt/grading/ca/mail.pem -noout -subject $ openssl verify -CAfile <(cat /opt/grading/ca/services.pem /opt/grading/ca/ca.pem) /opt/grading/ca/mail.pem |
| --- |
| **Expected Output**  > subject=CN = mail.dmz.worldskills.org  > /opt/grading/ca/mail.pem: OK |

# 10. Samba: jamie can login on Samba server

| **Command**  $ smbclient -L //localhost/ -I 127.0.0.1 -U "jamie%Skill39@Lyon" |
| --- |
| **Expected Output**  > Exit code was 0 |

# 11. Samba: /public is accessible without login

| **Command**  $ smbclient //localhost/public -I 127.0.0.1 -U "%" -c ls |
| --- |
| **Expected Output**  > lpcfg\_do\_global\_parameter: WARNING: The "encrypt passwords" option is deprecated  . D 0 Sun Jun 2 21:28:12 2024  .. D 0 Mon May 20 23:29:57 2024  lorem.txt A 12 Sun Jun 23 14:34:25 2024   31363900 blocks of size 1024. 24876808 blocks available |

# 12. Samba: /public is rw after login with a valid user

| **Command**  $ echo "Lorem Ipsum" > /tmp/lorem.txt; smbclient //localhost/public -I 127.0.0.1 -U "jamie%Skill39@Lyon" -c "put /tmp/lorem.txt lorem.txt" |
| --- |
| **Expected Output**  putting file /tmp/lorem.txt as \lorem.txt (11.7 kb/s) (average 11.7 kb/s) |

# 13. Samba: /public is ro without login

| **Command**  $ echo "Lorem Ipsum" > /tmp/lorem.txt; smbclient //localhost/public -I 127.0.0.1 -U "%" -c "put /tmp/lorem.txt lorem.txt" |
| --- |
| **Expected Output**  NT\_STATUS\_ACCESS\_DENIED opening remote file \lorem.txt |

# 14. Samba: /internal is not accessible without login

| **Command**  $ smbclient //localhost/internal -I 127.0.0.1 -U "%" -c "ls" 2>&1 || true |
| --- |
| **Expected Output**  tree connect failed: NT\_STATUS\_ACCESS\_DENIED |

# 15. Samba: /internal is rw after login

| **Command**  $ echo "Lorem Ipsum" > /tmp/lorem.txt; smbclient //localhost/internal -I 127.0.0.1 -U "jamie%Skill39@Lyon" -c "put /tmp/lorem.txt lorem.txt" |
| --- |
| **Expected Output**  putting file /tmp/lorem.txt as \lorem.txt (11.7 kb/s) (average 11.7 kb/s) |

# 16. DNS: A, AAAA and PTR for int-srv01.int.worldskills.org exists

| **Command**  $ dig +short +time=2 +tries=1 @127.0.0.1 int-srv01.int.worldskills.org. A $ dig +short +time=2 +tries=1 @127.0.0.1 -x 10.1.10.10 $ dig +short +time=2 +tries=1 @127.0.0.1 int-srv01.int.worldskills.org. AAAA $ dig +short +time=2 +tries=1 @127.0.0.1 -x 2001:db8:1001:10::10 |
| --- |
| **Expected Output**  > 10.1.10.10  > int-srv01.int.worldskills.org.  > 2001:db8:1001:10::10  > int-srv01.int.worldskills.org. |

# 17. DNS: SRV Record \_ldap.\_tcp.auth.int.worldskills.org exists

| **Command**  $ dig +short +time=2 +tries=1 @127.0.0.1 \_ldap.\_tcp.auth.int.worldskills.org SRV |
| --- |
| **Expected Output**  > 10 50 389 int-srv01.int.worldskills.org. |

# 18. DNS: Server accepts recursive queries

| **Command**  $ dig +recurse +time=2 +tries=1 @127.0.0.1 int.worldskills.org SOA |
| --- |
| **Expected Output**  Output does not contain: recursion requested but not available  > ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30658 ;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 |

# 19. DNS: int-srv01 is secondary Name server for dmz.worldskills.org

| **Command**  $ rndc zonestatus dmz.worldskills.org. $ dig +short +time=2 +tries=1 @127.0.0.1 dmz.worldskills.org SOA |
| --- |
| **Expected Output**  name: dmz.worldskills.org. type: secondary files: /var/cache/bind/dmz.worldskills.org.zone serial: 2023042601 nodes: 8 last loaded: Sun, 02 Jun 2024 20:13:03 GMT next refresh: Wed, 26 Jun 2024 23:08:51 GMT expires: Wed, 24 Jul 2024 00:21:29 GMT secure: no dynamic: no reconfigurable via modzone: no  > dmz.worldskills.org. admin.dmz.worldskills.org. 2023042601 86400 7200 2419200 3600 |

# 20. DNS: int-srv01 is secondary Name server for reverse zone 10.1.20.0/24 & 2001:db8:1001:20::/64

| **Command**  $ rndc zonestatus 20.1.10.in-addr.arpa. $ dig +short +time=2 +tries=1 @127.0.0.1 20.1.10.in-addr.arpa. SOA $ rndc zonestatus 0.2.0.0.1.0.0.1.8.b.d.0.1.0.0.2.ip6.arpa. $ dig +short +time=2 +tries=1 @127.0.0.1 0.2.0.0.1.0.0.1.8.b.d.0.1.0.0.2.ip6.arpa SOA |
| --- |
| **Expected Output**  name: 20.1.10.in-addr.arpa. type: secondary files: /var/cache/bind/20.1.10.in-addr.arpa.zone serial: 2023042701 nodes: 7 last loaded: Sun, 02 Jun 2024 16:53:48 GMT next refresh: Sun, 30 Jun 2024 03:07:46 GMT expires: Sat, 27 Jul 2024 08:51:31 GMT secure: no dynamic: no reconfigurable via modzone: no  Must contain dmz.worldskills.org > dmz.worldskills.org. admin.dmz.worldskills.org. 2023042601 86400 7200 2419200 3600  name: 0.2.0.0.1.0.0.1.8.b.d.0.1.0.0.2.ip6.arpa. type: secondary files: /var/cache/bind//0.2.0.0.1.0.0.1.8.b.d.0.1.0.0.2.ip6.arpa.zone serial: 2023042702 nodes: 11 last loaded: Sun, 02 Jun 2024 16:41:28 GMT next refresh: Sun, 30 Jun 2024 05:43:48 GMT expires: Sat, 27 Jul 2024 07:32:30 GMT secure: no dynamic: no reconfigurable via modzone: no  Must contain dmz.worldskills.org > dmz.worldskills.org. admin.dmz.worldskills.org. 2023042601 86400 7200 2419200 3600 |

# 21. IP Fowarding: IPv4 & IPv6 forwarding is enabled

| **Command**  $ sysctl -p |
| --- |
| **Expected Output**  > net.ipv4.ip\_forward = 1 > net.ipv6.conf.all.forwarding = 1 |

# 22. Firewall: Traffic to WAN uses masquerade NAT

| **Command**  $ timeout 2 bash -c "echo -e '\x1dclose\x0d' | telnet 1.1.1.20 22" |
| --- |
| **Expected Output**  > Trying 1.1.1.20... Connected to 1.1.1.20. Escape character is '^]'.  telnet> close Connection closed. |

# 23. WireGuard: Interface wg0 exists

| **Command**  $ wg show |
| --- |
| **Expected Output**  > interface: wg0  public key: ZIosm2a+SZqQPQ8L8dTMdLQjgWZGaDAGrxtgnVXzu3k=  private key: (hidden)  listening port: 1500  peer: RLsbxblPzbrTBJqh1oVnQQoyxNhLGw9H/gJJH322mjI=  preshared key: (hidden)  endpoint: 1.1.1.20:43226  allowed ips: 10.1.30.2/32, 2001:db8:1001:30::2/128  latest handshake: 1 minute, 35 seconds ago  transfer: 448.98 MiB received, 454.06 MiB sent |

# 24. WireGuard: file /etc/wireguard/wg0.conf contains a PSK

| **Command**  $ wg show wg0 preshared-keys |
| --- |
| **Expected Output**  > Exit code was 0 |

# 25. Transparent Proxy: A transparent proxy intercepts http traffic and adds the header x-secured-by: clearsky-proxy to the HTTP responses

| **Command**  $ curl -s -I --connect-timeout 3 http://10.1.20.10 2>&1 $ curl -s -I --connect-timeout 3 http://10.1.20.10 2>&1 $ curl -s -I --connect-timeout 3 http://[2001:db8:1001:20::20] 2>&1 $ curl -s -I --connect-timeout 3 http://[2001:db8:1001:20::20] 2>& |
| --- |
| **Expected Output**  4x HTTP requests in total (2x from jamie-ws01 & 2x from int-srv01) => IPv4 & IPv6 Response must contain ALWAYS > x-secured-by: clearsky-proxy  HTTP/1.1 301 Moved Permanently Content-Length: 0 Location: https://[2001:db8:1001:20::20]/ Date: Sat, 29 Jun 2024 15:05:08 GMT X-Cache: MISS from clearsky-proxy X-Cache-Lookup: MISS from clearsky-proxy:3128 Via: 1.1 clearsky-proxy (squid/5.7) Connection: keep-alive x-secured-by: clearsky-proxy |

# 26. Firewall: Firewall rules are well designed

| **Command**  nft list ruleset |
| --- |
| **Expected Output**  table ip nat {  chain prerouting {  type nat hook prerouting priority dstnat; policy accept;  iif "ens192" tcp dport 80 dnat to 10.1.20.20  iif "ens192" tcp dport 443 dnat to 10.1.20.20  iif "ens192" tcp dport 53 dnat to 10.1.20.20  iif "ens192" udp dport 53 dnat to 10.1.20.20  iifname { "wg0", "ens224" } tcp dport 80 redirect to :82  }   chain postrouting {  type nat hook postrouting priority srcnat; policy accept;  oif "ens192" masquerade  } } table ip filter {  chain input {  type filter hook input priority filter; policy drop;  iif "lo" accept  ct state established,related accept  iifname { "wg0", "ens224", "ens256" } accept  iifname "ens192" udp dport 1500 accept  icmp type { echo-reply, echo-request } limit rate 4/second accept  iifname "ens161" accept  }   chain forward {  type filter hook forward priority filter; policy drop;  ct state established,related accept  iifname { "ens224", "ens256" } oif "ens192" accept  iifname { "wg0", "ens224" } oif "ens256" accept  iifname "wg0" oif "ens224" accept  ip saddr 10.1.20.10 ip daddr 10.1.10.10 tcp dport 389 accept  ip daddr 10.1.20.20 tcp dport { 80, 443 } accept  ip daddr 10.1.20.20 tcp dport 53 accept  ip daddr 10.1.20.20 udp dport 53 accept  }   chain output {  type filter hook output priority filter; policy accept;  } } table ip6 nat {  chain prerouting {  type nat hook prerouting priority dstnat; policy accept;  iifname { "wg0", "ens224" } tcp dport 80 redirect to :82  } } table ip6 filter {  chain input {  type filter hook input priority filter; policy drop;  iif "lo" accept  ct state established,related accept  iifname { "wg0", "ens224", "ens256" } accept  iifname "wg0" oif "ens224" accept  ip6 daddr 2001:db8:1111::1 udp dport 1500 accept  ip6 daddr 2001:db8:1001:10::1 tcp dport 82 accept  }   chain forward {  type filter hook forward priority filter; policy drop;  ct state established,related accept  iifname { "wg0", "ens224", "ens256" } oif "ens192" accept  iifname { "wg0", "ens224" } oif "ens256" accept  iifname "wg0" oif "ens224" accept  ip6 daddr 2001:db8:1001:20::20 tcp dport { 80, 443 } accept  ip6 daddr 2001:db8:1001:20::20 tcp dport 53 accept  ip6 daddr 2001:db8:1001:20::20 udp dport 53 accept  }   chain output {  type filter hook output priority filter; policy accept;  } } |

# 27. Mail Server: LDAP user jamie can login and send an email to himself

| **Command**  printf 'Subject: WSC2024\_FLAG\n\nWSC2024\_FLAG' | curl -s -k --ssl-reqd --url smtps://localhost --user jamie:Skill39@Lyon -H 'Subject: WSC2024\_FLAG' --mail-from 'jamie.oliver@dmz.worldskills.org(Jamie Oliver)' --mail-rcpt jamie.oliver@dmz.worldskills.org --upload-file - OR printf 'Subject: WSC2024\_FLAG\n\nWSC2024\_FLAG' | curl -s -k --url smtp://localhost --user jamie:Skill39@Lyon -H 'Subject: WSC2024\_FLAG' --mail-from 'jamie.oliver@dmz.worldskills.org(Jamie Oliver)' --mail-rcpt jamie.oliver@dmz.worldskills.org --upload-file - IFS=" " read -r -a mail\_ids <<< "$(curl -s -k --user jamie:Skill39@Lyon imaps://localhost/INBOX?SUBJECT%20WSC2024\_FLAG 2>&1 | grep -oP '(?<=SEARCH)[ 0-9]+')" && curl -s -k --user jamie:Skill39@Lyon "imaps://localhost/INBOX;MAILINDEX=${mail\_ids[-1]}" 2>&1 OR IFS=" " read -r -a mail\_ids <<< "$(curl -s -k --user jamie:Skill39@Lyon imap://localhost/INBOX?SUBJECT%20WSC2024\_FLAG 2>&1 | grep -oP '(?<=SEARCH)[ 0-9]+')" && curl -s -k --user jamie:Skill39@Lyon "imap://localhost/INBOX;MAILINDEX=${mail\_ids[-1]}" 2>&1 |
| --- |
| **Expected Output**  Return-Path: <jamie.oliver@dmz.worldskills.org> Delivered-To: jamie.oliver@dmz.worldskills.org Received: from mail (localhost [127.0.0.1])  (using TLSv1.3 with cipher TLS\_AES\_256\_GCM\_SHA384 (256/256 bits)  key-exchange X25519 server-signature RSA-PSS (4096 bits) server-digest SHA256)  (No client certificate requested)  by mail.dmz.worldskills.org (Postfix) with ESMTPSA id 211E940BB0  for <jamie.oliver@dmz.worldskills.org>; Sat, 29 Jun 2024 17:07:42 +0200 (CEST) Subject: WSC2024\_FLAG Message-Id: <20240629150742.211E940BB0@mail.dmz.worldskills.org> Date: Sat, 29 Jun 2024 17:07:42 +0200 (CEST) From: jamie.oliver@dmz.worldskills.org  WSC2024\_FLAG |

# 28. Mail Server: echo@dmz.worldskills.org auto-replies/echos email (no NDR!)

| **Command**  $ printf 'Subject: WSC2024\_ECHO\_FLAG\nFrom: jamie.oliver@dmz.worldskills.org\nTo: echo@dmz.worldskills.org\n\nWSC2024\_ECHO\_FLAG' | curl -s -k --ssl-reqd --url smtps://localhost --user jamie:Skill39@Lyon --mail-from 'jamie.oliver@dmz.worldskills.org(Jamie Oliver)' --mail-rcpt echo@dmz.worldskills.org --upload-file - OR $ printf 'Subject: WSC2024\_ECHO\_FLAG\nFrom: jamie.oliver@dmz.worldskills.org\nTo: echo@dmz.worldskills.org\n\nWSC2024\_ECHO\_FLAG' | curl -s -k --url smtp://localhost --user jamie:Skill39@Lyon --mail-from 'jamie.oliver@dmz.worldskills.org(Jamie Oliver)' --mail-rcpt echo@dmz.worldskills.org --upload-file - $ IFS=" " read -r -a mail\_ids <<< "$(curl -s -k --user jamie:Skill39@Lyon imaps://localhost/INBOX?FROM%20echo@dmz.worldskills.org 2>&1 | grep -oP '(?<=SEARCH)[ 0-9]+')" && curl -s -k --user jamie:Skill39@Lyon "imaps://localhost/INBOX;MAILINDEX=${mail\_ids[-1]}" 2>&1 OR $ IFS=" " read -r -a mail\_ids <<< "$(curl -s -k --user jamie:Skill39@Lyon imap://localhost/INBOX?FROM%20echo@dmz.worldskills.org 2>&1 | grep -oP '(?<=SEARCH)[ 0-9]+')" && curl -s -k --user jamie:Skill39@Lyon "imap://localhost/INBOX;MAILINDEX=${mail\_ids[-1]}" 2>&1 |
| --- |
| **Expected Output**  Message contains WSC2024\_ECHO\_FLAG or a custom message and following strings as well:  Return-Path: <jamie.oliver@dmz.worldskills.org> Delivered-To: jamie.oliver@dmz.worldskills.org Received: by mail.dmz.worldskills.org (Postfix, from userid 1000)  id 278FD40BBE; Sat, 29 Jun 2024 17:16:55 +0200 (CEST) X-Sieve: Pigeonhole Sieve 0.5.19 (4eae2f79) X-Sieve-Redirected-From: echo@dmz.worldskills.org Subject: Echo: WSC2024\_ECHO\_FLAG To: jamie.oliver@dmz.worldskills.org From: echo@dmz.worldskills.org Delivered-To: echo@dmz.worldskills.org Received: from mail (localhost [127.0.0.1])  (using TLSv1.3 with cipher TLS\_AES\_256\_GCM\_SHA384 (256/256 bits)  key-exchange X25519 server-signature RSA-PSS (4096 bits) server-digest SHA256)  (No client certificate requested)  by mail.dmz.worldskills.org (Postfix) with ESMTPSA id 1FD3340BAB  for <echo@dmz.worldskills.org>; Sat, 29 Jun 2024 17:16:55 +0200 (CEST) Subject: WSC2024\_ECHO\_FLAG Message-Id: <20240629151655.1FD3340BAB@mail.dmz.worldskills.org> Date: Sat, 29 Jun 2024 17:16:55 +0200 (CEST)  WSC2024\_ECHO\_FLAG |

# 29. Backup: /dev/sdb has been added to /etc/fstab

| **Command**  $ cat /etc/fstab |
| --- |
| **Expected Output**  > # /etc/fstab: static file system information. # # Use 'blkid' to print the universally unique identifier for a # device; this may be used with UUID= as a more robust way to name devices # that works even if disks are added and removed. See fstab(5). # # systemd generates mount units based on this file, see systemd.mount(5). # Please run 'systemctl daemon-reload' after making changes here. # # <file system> <mount point> <type> <options> <dump> <pass> /dev/mapper/debian--vg-root / ext4 errors=remount-ro 0 1 # /boot was on /dev/sda2 during installation UUID=567cbd87-0262-4e06-b19b-68a69dd383fc /boot ext2 defaults 0 2 /dev/mapper/debian--vg-swap\_1 none swap sw 0 0 /dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0 /dev/sr1 /media/cdrom1 udf,iso9660 user,noauto 0 0 /dev/sr2 /media/cdrom2 udf,iso9660 user,noauto 0 0 /dev/sr3 /media/cdrom3 udf,iso9660 user,noauto 0 0 /dev/sr4 /media/cdrom4 udf,iso9660 user,noauto 0 0 UUID=485f4ce8-1f70-4c31-a2ab-4a5fb9602f56 /opt/backup ext4 defaults 0 0 |

# 30. SSH: Openssh option TrustedUserCAKeys is populated

| **Command**  $ sshd -T |
| --- |
| **Expected Output**  > trustedusercakeys /etc/ssh/ca.key.pub |

# 31. SSH: user root from ha-prx01 can ssh as root using the signed key

| **Command**  $ timeout 2 bash -c 'ssh -vv -o StrictHostKeyChecking=no root@10.1.20.10 "lsb\_release -is" 2>&1 | grep "Server accepts"' |
| --- |
| **Expected Output**  > debug1: Server accepts key: /root/.ssh/root.key-cert.pub ED25519-CERT SHA256:utUmDuUa+2a+I3G/lqCBsMM+MHa9LCqXMrh0GFH34Yg explicit |

# 32. Backup: Backup script style

| **Command**  $ rm -rf /opt/backup/\* ; bash /opt/backup.sh $ find /opt/backup/ -name dovecot.conf -o -name main.cf -o -name dovecot.index\* |
| --- |
| **Expected Output**  > /opt/backup/mailboxes/mailboxes/jamie/Maildir/dovecot.index.cache /opt/backup/dovecot/dovecot/dovecot.conf /opt/backup/postfix/postfix/main.cf |

# 33. Reverse Proxy: Services is listening on Port 80/tcp (HTTP) (IPv4 + IPv6)

| **Command**  $ curl -vs --connect-timeout 2 http://127.0.0.1/ 2>&1 ; curl -vs --connect-timeout 2 http://[::1]/ 2>&1 $ curl -vs --connect-timeout 2 http://127.0.0.1/ 2>&1 ; curl -vs --connect-timeout 2 http://[::1]/ 2>&1 |
| --- |
| **Expected Output**  ha-prx01 & ha-prx02 have following output > \* Trying 127.0.0.1:80... \* Connected to 127.0.0.1 (127.0.0.1) port 80 (#0) > GET / HTTP/1.1 > Host: 127.0.0.1 > User-Agent: curl/7.88.1 > Accept: \*/\* >  < HTTP/1.1 301 Moved Permanently < content-length: 0 < location: https://127.0.0.1/ <  \* Connection #0 to host 127.0.0.1 left intact \* Trying [::1]:80... \* Connected to ::1 (::1) port 80 (#0) > GET / HTTP/1.1 > Host: [::1] > User-Agent: curl/7.88.1 > Accept: \*/\* >  < HTTP/1.1 301 Moved Permanently < content-length: 0 < location: https://[::1]/ <  \* Connection #0 to host ::1 left intact |

# 34. Reverse Proxy: Services is listening on Port 443/tcp (HTTPS) (IPv4 + IPv6)

| **Command**  $ curl -kvs --connect-timeout 2 https://127.0.0.1/ 2>&1 ; curl -kvs --connect-timeout 2 https://[::1]/ 2>&1 $ curl -kvs --connect-timeout 2 https://127.0.0.1/ 2>&1 ; curl -kvs --connect-timeout 2 https://[::1]/ 2>&1 |
| --- |
| **Expected Output**  ha-prx01 & ha-prx02 have following output > \* Connected to 127.0.0.1 (127.0.0.1) port 443 (#0)  > \* Connected to ::1 (::1) port 443 (#0) |

# 35. Reverse Proxy: Services is proxying the webservers

| **Command**  Run 4x the same command on mail => $ curl -ks --connect-timeout 2 https://www.dmz.worldskills.org/whoami 2>&1 |
| --- |
| **Expected Output**  at least two different values e.g. > web01 > web02 |

# 36. Reverse Proxy: adds the header "via-proxy: ha-prx01" to the http responses

| **Command**  $ curl -kvs --connect-timeout 2 https://www.dmz.worldskills.org/whoami 2>&1 |
| --- |
| **Expected Output**  via-proxy with the hostname of one of the reverse proxy exists. E.g.  < HTTP/1.1 200 OK < server: nginx < date: Sat, 29 Jun 2024 15:27:35 GMT < content-type: text/plain; charset=utf-8 < content-length: 5 < via-proxy: ha-prx02 <  { [5 bytes data] \* Connection #0 to host www.dmz.worldskills.org left intact web01 |

# 37. Reverse Proxy: HTTP requests get redirected to HTTPS

| **Command**  $ curl -kvs -I --connect-timeout 2 http://www.dmz.worldskills.org/ 2>&1 |
| --- |
| **Expected Output**  > \* Trying [2001:db8:1001:20::20]:80... \* Connected to www.dmz.worldskills.org (2001:db8:1001:20::20) port 80 (#0) > HEAD / HTTP/1.1 > Host: www.dmz.worldskills.org > User-Agent: curl/7.88.1 > Accept: \*/\* >  < HTTP/1.1 301 Moved Permanently < content-length: 0 < location: https://www.dmz.worldskills.org/ <  \* Connection #0 to host www.dmz.worldskills.org left intact HTTP/1.1 301 Moved Permanently content-length: 0 location: https://www.dmz.worldskills.org/ |

# 38. Reverse Proxy: HTTPS certificate is signed by CA

| **Command**  scp /opt/grading/ca/ca.pem 10.1.20.10:/tmp/ca.pem $ timeout 2 bash -c 'echo "Q" | openssl s\_client -connect www.dmz.worldskills.org:443 -CAfile /tmp/ca.pem' |
| --- |
| **Expected Output**  CONNECTED(00000003) --- Certificate chain  0 s:CN = www.dmz.worldskills.org  i:CN = ClearSky Services CA  a:PKEY: rsaEncryption, 4096 (bit); sigalg: RSA-SHA256  v:NotBefore: May 20 21:30:13 2024 GMT; NotAfter: May 18 21:30:13 2034 GMT  1 s:CN = ClearSky Services CA  i:CN = ClearSky Root CA  a:PKEY: rsaEncryption, 4096 (bit); sigalg: RSA-SHA256  v:NotBefore: May 20 21:30:12 2024 GMT; NotAfter: May 18 21:30:12 2034 GMT .... SSL handshake has read 3485 bytes and written 409 bytes Verification: OK --- New, TLSv1.3, Cipher is TLS\_AES\_256\_GCM\_SHA384 Server public key is 4096 bit |

# 39. Reverse Proxy: All services are available on the HA-IP 10.1.20.20/24 / 2001:db8:1001:20::20/64 and are highly available (failover works)

| **Command**  $ curl -4vks --connect-timeout 2 https://www.dmz.worldskills.org 2>&1 ; curl -6vks --connect-timeout 2 https://www.dmz.worldskills.org 2>&1 |
| --- |
| **Expected Output**  > \* Connected to www.dmz.worldskills.org (10.1.20.20) port 443 (#0) > \* Connected to www.dmz.worldskills.org (2001:db8:1001:20::20) port 443 (#0) |

# 40. DNS: ha-prx01 is primary for dmz.worldskills.org and ha-prx02 is secondary

| **Command**  $ rndc zonestatus dmz.worldskills.org. $ rndc zonestatus dmz.worldskills.org. |
| --- |
| **Expected Output**  ha-prx01 > name: dmz.worldskills.org. type: primary files: /etc/bind/dmz.worldskills.org.zone serial: 2023042601 nodes: 8 last loaded: Sat, 08 Jun 2024 15:52:14 GMT secure: no dynamic: no reconfigurable via modzone: no  ha-prx02 > name: dmz.worldskills.org. type: secondary files: /var/cache/bind/dmz.worldskills.org.zone serial: 2023042601 nodes: 8 next refresh: Sat, 29 Jun 2024 20:12:29 GMT expires: Fri, 26 Jul 2024 23:25:36 GMT secure: no dynamic: no reconfigurable via modzone: no |

# 41. DNS: Server denies recursive queries

| **Command**  $ dig +recurse +time=2 +tries=1 @127.0.0.1 int.worldskills.org SOA |
| --- |
| **Expected Output**  <<>> DiG 9.18.19-1~deb12u1-Debian <<>> +recurse +time +tries @127.0.0.1 int.worldskills.org SOA ; (1 server found) ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 6382 ;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1 ;; WARNING: recursion requested but not available |

# 42. DNS: A & PTR Record for all hosts in DMZ exists (mail.dmz.worldskills.org, ha-prx01.dmz.worldskills.org, ha-prx02.dmz.worldskills.org, web01.dmz.worldskills.org, web02.dmz.worldskills.org)

| **Command**  > $ dig +short +time=2 +tries=1 @127.0.0.1 mail.dmz.worldskills.org. A $ dig +short +time=2 +tries=1 @127.0.0.1 -x 10.1.20.10 $ dig +short +time=2 +tries=1 @127.0.0.1 ha-prx01.dmz.worldskills.org. A $ dig +short +time=2 +tries=1 @127.0.0.1 -x 10.1.20.21 $ dig +short +time=2 +tries=1 @127.0.0.1 ha-prx02.dmz.worldskills.org. A $ dig +short +time=2 +tries=1 @127.0.0.1 -x 10.1.20.22 $ dig +short +time=2 +tries=1 @127.0.0.1 web01.dmz.worldskills.org. A $ dig +short +time=2 +tries=1 @127.0.0.1 -x 10.1.20.31 $ dig +short +time=2 +tries=1 @127.0.0.1 web02.dmz.worldskills.org. A $ dig +short +time=2 +tries=1 @127.0.0.1 -x 10.1.20.32 |
| --- |
| **Expected Output**  > $ dig +short +time=2 +tries=1 @127.0.0.1 mail.dmz.worldskills.org. A 10.1.20.10  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 -x 10.1.20.10 mail.dmz.worldskills.org.  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 ha-prx01.dmz.worldskills.org. A 10.1.20.21  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 -x 10.1.20.21 ha-prx01.dmz.worldskills.org.  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 ha-prx02.dmz.worldskills.org. A 10.1.20.22  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 -x 10.1.20.22 ha-prx02.dmz.worldskills.org.  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 web01.dmz.worldskills.org. A 10.1.20.31  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 -x 10.1.20.31 web01.dmz.worldskills.org.  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 web02.dmz.worldskills.org. A 10.1.20.32  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 -x 10.1.20.32 web02.dmz.worldskills.org. |

# 43. DNS: A, AAAA & PTR Record prx-vrrp.dmz.worldskills.org exists

| **Command**  $ dig +short +time=2 +tries=1 @127.0.0.1 prx-vrrp.dmz.worldskills.org. A $ dig +short +time=2 +tries=1 @127.0.0.1 -x 10.1.20.20 $ dig +short +time=2 +tries=1 @127.0.0.1 prx-vrrp.dmz.worldskills.org. AAAA $ dig +short +time=2 +tries=1 @127.0.0.1 -x 2001:db8:1001:20::20 |
| --- |
| **Expected Output**  > $ dig +short +time=2 +tries=1 @127.0.0.1 prx-vrrp.dmz.worldskills.org. A 10.1.20.20  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 -x 10.1.20.20 prx-vrrp.dmz.worldskills.org.  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 prx-vrrp.dmz.worldskills.org. AAAA 2001:db8:1001:20::20  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 -x 2001:db8:1001:20::20 prx-vrrp.dmz.worldskills.org. |

# 44. DNS: AAAA & PTR Record for all hosts in DMZ exists (mail.dmz.worldskills.org, ha-prx01.dmz.worldskills.org, ha-prx02.dmz.worldskills.org, web01.dmz.worldskills.org, web02.dmz.worldskills.org)

| **Command**  $ dig +short +time=2 +tries=1 @127.0.0.1 mail.dmz.worldskills.org. AAAA $ dig +short +time=2 +tries=1 @127.0.0.1 -x 2001:db8:1001:20::10 $ dig +short +time=2 +tries=1 @127.0.0.1 ha-prx01.dmz.worldskills.org. AAAA $ dig +short +time=2 +tries=1 @127.0.0.1 -x 2001:db8:1001:20::21 $ dig +short +time=2 +tries=1 @127.0.0.1 ha-prx02.dmz.worldskills.org. AAAA $ dig +short +time=2 +tries=1 @127.0.0.1 -x 2001:db8:1001:20::22 $ dig +short +time=2 +tries=1 @127.0.0.1 web01.dmz.worldskills.org. AAAA $ dig +short +time=2 +tries=1 @127.0.0.1 -x 2001:db8:1001:20::31 $ dig +short +time=2 +tries=1 @127.0.0.1 web02.dmz.worldskills.org. AAAA $ dig +short +time=2 +tries=1 @127.0.0.1 -x 2001:db8:1001:20::32 |
| --- |
| **Expected Output**  > $ dig +short +time=2 +tries=1 @127.0.0.1 mail.dmz.worldskills.org. AAAA 2001:db8:1001:20::10  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 -x 2001:db8:1001:20::10 mail.dmz.worldskills.org.  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 ha-prx01.dmz.worldskills.org. AAAA 2001:db8:1001:20::21  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 -x 2001:db8:1001:20::21 ha-prx01.dmz.worldskills.org.  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 ha-prx02.dmz.worldskills.org. AAAA 2001:db8:1001:20::22  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 -x 2001:db8:1001:20::22 ha-prx02.dmz.worldskills.org.  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 web01.dmz.worldskills.org. AAAA 2001:db8:1001:20::31  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 -x 2001:db8:1001:20::31 web01.dmz.worldskills.org.  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 web02.dmz.worldskills.org. AAAA 2001:db8:1001:20::32  Executed command on ha-prx01 => $ dig +short +time=2 +tries=1 @127.0.0.1 -x 2001:db8:1001:20::32 web02.dmz.worldskills.org. |

# 45. DNS: CNAME Record www.dmz.worldskills.org exists and points to prx-vrrp.dmz.worldskills.org

| **Command**  $ dig +short +time=2 +tries=1 @127.0.0.1 www.dmz.worldskills.org. CNAME |
| --- |
| **Expected Output**  > prx-vrrp.dmz.worldskills.org. |

# 46. Web Server: main.html is served when the root page is opened

| **Command**  $ curl -s --connect-timeout 2 http://127.0.0.1/ 2>&1 |
| --- |
| **Expected Output**  web01 print the following output: > ID=7ecb4e033421b96f88e2111fe97cdad4ddceacc6 |

# 47. Web Server: 404.html is served when an invalid path is given

| **Command**  $ curl -s --connect-timeout 2 http://127.0.0.1/loremipsum 2>&1 |
| --- |
| **Expected Output**  web01 print the following output: > Sorry, but something went wrong |

# 48. Port forwarding WAN>prx-vrrp port 53/tcp/udp exists

| **Command**  $ dig +short +time=2 +tries=1 @1.1.1.10 dmz.worldskills.org SOA $ dig +tcp +short +time=2 +tries=1 @1.1.1.10 dmz.worldskills.org SOA |
| --- |
| **Expected Output**  > dmz.worldskills.org. admin.dmz.worldskills.org. 2023042601 86400 7200 2419200 3600  > dmz.worldskills.org. admin.dmz.worldskills.org. 2023042601 86400 7200 2419200 3600 |

# 49. Port forwarding WAN>prx-vrrp port 80/tcp and port 443/tcp exist

| **Command**  $ curl -sv --connect-timeout 2 http://1.1.1.10  $ curl -ksv --connect-timeout 2 https://1.1.1.10 |
| --- |
| **Expected Output**  > \* Trying 1.1.1.10:80... \* Connected to 1.1.1.10 (1.1.1.10) port 80 (#0) > GET / HTTP/1.1 > Host: 1.1.1.10 > User-Agent: curl/7.88.1 > Accept: \*/\* >  < HTTP/1.1 301 Moved Permanently < content-length: 0 < location: https://1.1.1.10/ <  \* Connection #0 to host 1.1.1.10 left intact ... \* Trying 1.1.1.10:443... \* Connected to 1.1.1.10 (1.1.1.10) port 443 (#0) |

# 50. www.dmz.worldskills.org is defined as startpage, can be reached and doesn't show any certificate errors

| **Command**  Open firefox on jamie-ws01 |
| --- |
| **Expected Output**  www.dmz.worldskills.org is shown without any certificate errors |

# 51. Service IMAP allows STARTTLS and uses certificate signed by CA

| **Command**  scp /opt/grading/ca/ca.pem 10.1.20.10:/tmp/ca.pem $ timeout 2 bash -c 'echo "Q" | openssl s\_client -connect 10.1.20.10:143 -verify\_return\_error -starttls imap -CAfile /tmp/ca.pem' |
| --- |
| **Expected Output**  depth=2 CN = ClearSky Root CA verify return:1 depth=1 CN = ClearSky Services CA verify return:1 depth=0 CN = mail.dmz.worldskills.org verify return:1 CONNECTED(00000003) --- Certificate chain  0 s:CN = mail.dmz.worldskills.org  i:CN = ClearSky Services CA  a:PKEY: rsaEncryption, 4096 (bit); sigalg: RSA-SHA256  v:NotBefore: May 5 20:30:07 2024 GMT; NotAfter: May 3 20:30:07 2034 GMT  1 s:CN = ClearSky Services CA  i:CN = ClearSky Root CA  a:PKEY: rsaEncryption, 4096 (bit); sigalg: RSA-SHA256  v:NotBefore: May 5 20:30:06 2024 GMT; NotAfter: May 3 20:30:06 2034 GMT <CERTIFICATE> subject=CN = mail.dmz.worldskills.org issuer=CN = ClearSky Services CA --- No client certificate CA names sent Peer signing digest: SHA256 Peer signature type: RSA-PSS Server Temp Key: X25519, 253 bits --- SSL handshake has read 3807 bytes and written 403 bytes Verification: OK --- New, TLSv1.3, Cipher is TLS\_AES\_256\_GCM\_SHA384 Server public key is 4096 bit Secure Renegotiation IS NOT supported Compression: NONE Expansion: NONE No ALPN negotiated Early data was not sent Verify return code: 0 (ok) --- . OK Pre-login capabilities listed, post-login capabilities have more. DONE |

# 52. Thunderbird is setup with the mailbox from jamie

| **Command**  Open thunderbird on jamie-ws01 |
| --- |
| **Expected Output**  mailbox from jamie is set up Send a test email to jamie.oliver@dmz.worldskills.org using Thunderbird |

# 53. wg is configured to allow both IPv4 and IPv6

| **Command**  $ wg show interfaces $ (ip route get 10.1.10.10 ; ip -6 route get 2001:db8:1001:10::10) | grep <interface\_name> | wc -l |
| --- |
| **Expected Output**  any interface is allowed (e.g. clearsky). But the count must be 2  > 2 |

# 54. jamie-ws01 can access SSH on int-srv01 via VPN

| **Command**  $ dig +time=2 +tries=1 +short @10.1.10.10 int.worldskills.org SOA $ dig +time=2 +tries=1 +short @2001:db8:1001:10::10 int.worldskills.org SOA |
| --- |
| **Expected Output**  > int.worldskills.org. admin.dmz.worldskills.org. 2023042601 86400 7200 2419200 3600  > int.worldskills.org. admin.dmz.worldskills.org. 2023042601 86400 7200 2419200 3600 |

# 55. Ansible can connect to web02

| **Command**  cd /opt/ansible/ && timeout 10 ansible -m ping all |
| --- |
| **Expected Output**  web02 | SUCCESS => {  "ansible\_facts": {  "discovered\_interpreter\_python": "/usr/bin/python3"  },  "changed": false,  "ping": "pong" } |

# 56. Webserver serves required web pages (index, 404, whoami)

| **Command**  $ curl -s --connect-timeout 2 http://127.0.0.1/ 2>&1 $ curl -s --connect-timeout 2 http://127.0.0.1/invalid 2>&1 $ curl -s --connect-timeout 2 http://127.0.0.1/whoami 2>&1 |
| --- |
| **Expected Output**  The output:from web02 contains following strings: > ID=7ecb4e033421b96f88e2111fe97cdad4ddceacc6 > Sorry, but something went wrong > web02 |