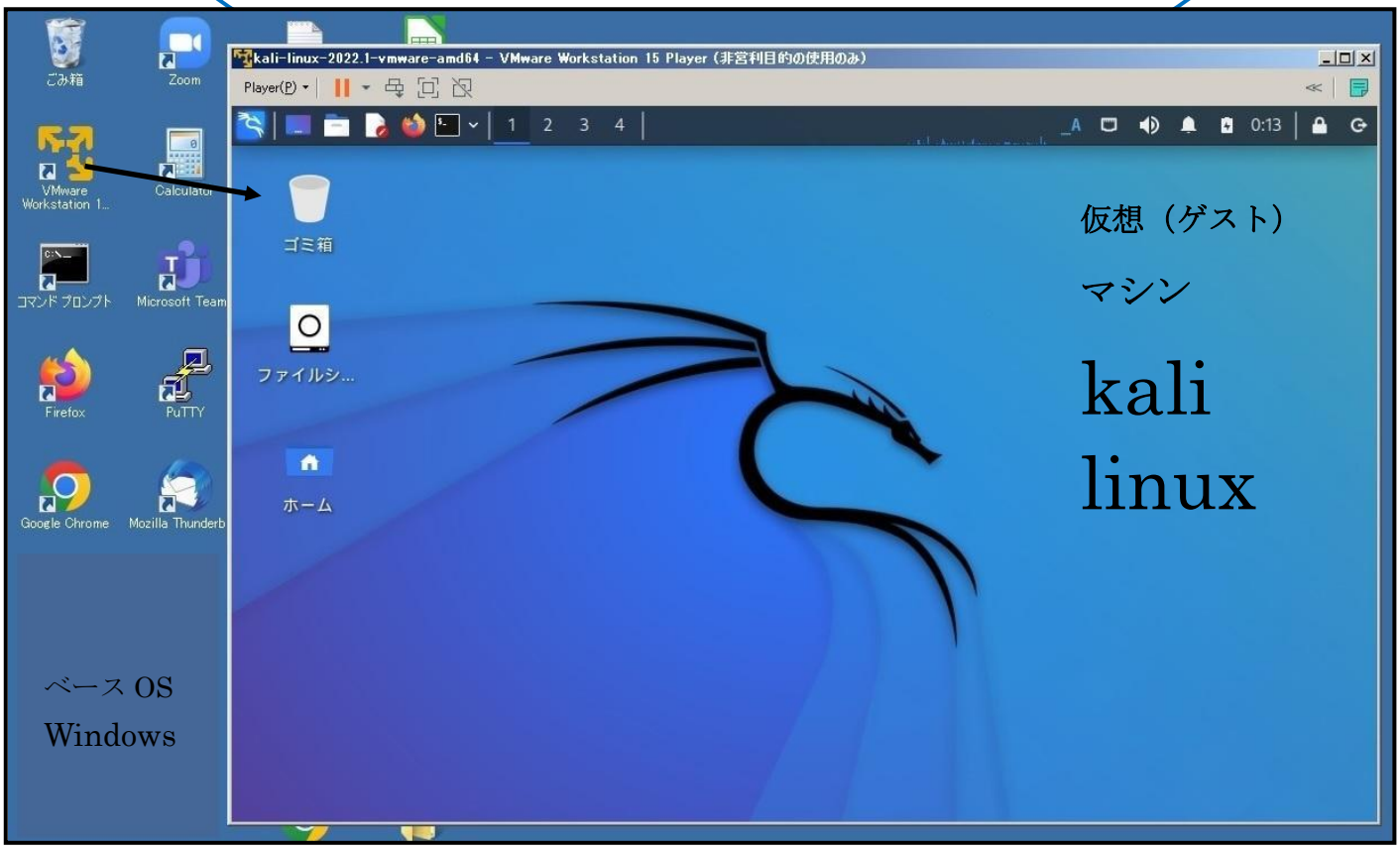
コンピュータ科学（第２４話）

第２４話（セキュリティⅢ：矛）　ネットワークハッキング

タヌキ、第２２話と第２３話は防御（盾）のセキュリティの話だったが、これからは攻め（矛）のセキュリティの話だ。
第８話から第１２話で話した仮想マシンを使えばハッキングの効果的な実習ができるぞ。仮想マシンに「kali linux」を使うのだ。
「kali linux」をハッキングするサーバに見立て、ベース OS の Windows を攻撃をしかける端末と考える。こうすれば、１台のノートパソコンでどこでもハッキングの勉強ができるぞ。さらに「kali linux」にはハッキング実習の為の多くのツール（aircrack-ng、burpsuit、hyda、john、nmap、sqlmap、wireshark など）が標準でインストールされている。
前にも言ったように、ハッキングの学習にインターネット上の実在するサーバを対象にすることはできないのだ。これは守ってくれよな。
それで、仮想環境は技術を身に付ける上で非常に重要なんだ。仮想環境さえあれば、「kali-linux-2022.1-vmware-amd64.7z」をダウンロードし、「7 z 2107-x64」で解凍し、適当なフォルダに保存し、保存先を指定して「仮想マシンを開く」というボタンをクリックするだけで、以下のような画像が表示され使えるようになるよ。つまり、解凍した「kali-linux-2022.1-・」自体が仮想マシンなのだ。



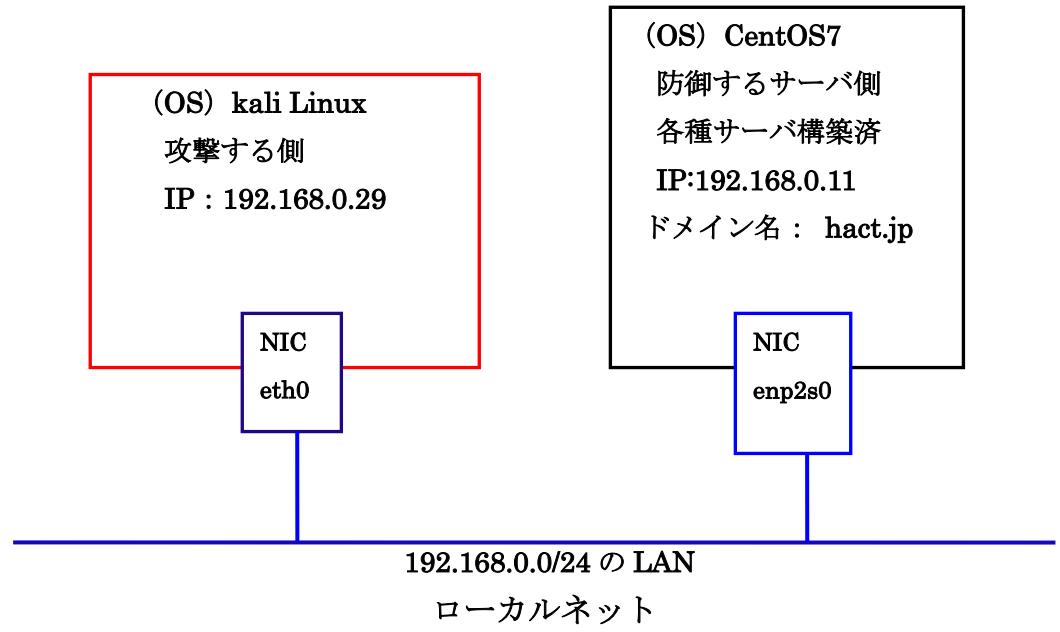
キツネ！オイラも第８話から第１２話で作っておいた VMware の仮想環境で「kali linux」を起動してみた。簡単に起動できたが、日本語が使えるようにするのが大変だった。当然ネットを参考にしてできたがね。
でも、「kali linux」ってカッコええね！これを見ているだけでホワイトハッカーになれそうな気がするわ。

確かに「kali linux」ってカッコええよな。
ただ、「kali linux」は、Debian（Ubuntu）系の Linux だから、Ubuntu のコマンドの使い方に馴れていれば操作が楽だよ。
さて、ネットワークハッキングは、ネットワーク上のサーバへの攻撃だ。ホワイトハッカーの場合は、何故そのサーバを攻撃するのか、しなければならないのか、という大義名分が必要だ。
大義名分があってこそ、攻撃する目的が明確になる。例を挙げると以下のようになる。
①サーバのルート権限を取得し、サーバ全体をコントロールする。
②サーバ内の特定の情報だけを取得する。
③他のユーザが特定のサーバにアクセスできないようにする。
などだ。目的によって手法が異なってくるのだ。

なるほど、大義名分を持たずに闇雲にハッキングすると愉快犯かブラックハッカーになってしまうわけか。
オイラも注意しなくちゃな。

ハッキングの実習をする為には、それなりの学習環境が必要だ。それを以下に示しておく。前にも言ったように、ハッキングする対象のサーバがどうしても必要だ。それに、攻撃する側の端末も必要だ。その端末を攻撃用の豊富なツールを備えた「kali Linux」にした。

［ネットワーク・ハッキング実習の為の学習環境］



確かに、この程度の実習環境を構築できない学生には、ネットワーク・ハッキングの学習は無理か。オイラも頑張って作るぞ。

とにかくネットワークハッキングをする為には、ネットワークに関する基本的な知識が必要だ。先ず、ネットワーク上の PC の接続を確認する「ifconfig(ipconfig : windows の場合)、ping、traceroute(tracert : windows の場合)、nslookup」などは基本的なコマンドなので、使ってみる必要があるよ。
では、先ずターゲットサーバがネットワーク上に存在するか判定する ping コマンドからテストしてみるか。攻撃側は、「kali linux」、ターゲットは CentOS7 サーバだ。「kali linux」側から ping を送ってみよう。ping には、「-a –c –f –i -j –k –l –n –r –s –t –v -w」などという複数のオプションがあるから必要に応じて使い分けてくれ。守りの CentOS 7 側では ping を受け付けないようにフィルタリング設定することもできるんだ。以下の画面は受け付けられた時の画面だ。

```
ファイル  操作  編集  表示  ヘルプ
└─# ping -c 3  192.168.0.11
PING 192.168.0.11 (192.168.0.11) 56(84) bytes of data.
64 bytes from 192.168.0.11: icmp_seq=1 ttl=64 time=0.641 ms
64 bytes from 192.168.0.11: icmp_seq=2 ttl=64 time=0.386 ms
64 bytes from 192.168.0.11: icmp_seq=3 ttl=64 time=0.504 ms

── 192.168.0.11 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2033ms
rtt min/avg/max/mdev = 0.386/0.510/0.641/0.104 ms
```

CentOS7 サーバが存在することが確認できたね。オプション「-c　3」は、３回発信してみる、ということか。
次は、どんなコマンドだ！

自分のネットワーク環境（NIC の状態）も知らなくて
はならないから、ifconfig も使ってみよう。
以下が、その表示画面だ。

```
┌──(root㉿kali)-[/home/kali]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.29  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::20c:29ff:fe7f:57d  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:7f:05:7d  txqueuelen 1000  （イーサネット）
        RX packets 12254  bytes 784549 (766.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2117  bytes 180158 (175.9 KiB)
        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
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  （ローカルループバック）
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

ifconfig を使うと NIC に割り振られたデバイス名（eth0）も
わかるんだ。それ以外にも色々とわかることがあるのだ。
デバイス名（lo）は、ローカルエリアネットワークか。
キツネ、次は。

次は、DNS サーバの存在を確認する nslookup だ。
正引きと逆引きがあるからね。

```
┌──(root㉿kali)-[/home/kali]
└─# nslookup
> 192.168.0.11
11.0.168.192.in-addr.arpa       name = ns.hact.jp.
> www.hact.jp
Server:         192.168.0.11
Address:        192.168.0.11#53

www.hact.jp     canonical name = ns.hact.jp.
Name:   ns.hact.jp
Address: 192.168.0.11
>
```

逆引き

正引き

逆引きでドメイン名（hact.jp）を知った上で host コマンドを使うと、DNS サーバの正引きゾーンデータベースファイルの記述内容を見ることができるよ。
これをどう使うかは、ハッキングの知識とタヌキ自身が持つ独創性によるよ。ハッキングのツールを学び、使い方を身に付けることはできるが、程々のハッカーならば、知っていることだ。課題は、そのツール類を使って他のハッカーが思いつかない活用方法を見つけることだ。ハッカー個々人が持っている独創性によって優劣が決まるのさ。

```
┌──(root㉿kali)-[/home/kali]
└─# host -a hact.jp
Trying "hact.jp"
;; ⟶⟶HEADER⟵⟵ opcode: QUERY, status: NOERROR, id: 60552
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; QUESTION SECTION:
;hact.jp.                       IN      ANY

;; ANSWER SECTION:
hact.jp.                86400   IN      SOA     ns.hact.jp. root.hact.jp. 202
2051602 3600 1800 604800 86400
hact.jp.                86400   IN      NS      ns.hact.jp.

;; ADDITIONAL SECTION:
ns.hact.jp.             86400   IN      A       192.168.0.11

Received 99 bytes from 192.168.0.11#53 in 11 ms
```

なるほどな、確かに DNS サーバのゾーンデータベースファイルだが、これを見てどのように使うかな。やはり、ROOT 権限がほしいよな。

そうだよな、ROOT 権限をどのように取得するかですよね。これ以上の話は、パケットの分析などが入ってくるので、次回、第２５話に譲ることにする。

どのように展開されるか、後、御期待だ！