*SSH*epherd

Removing the SSH Attack Surface

A critical component to a multi-layered security strategy

What is *SSH*epherd?

- A software security product
- Removes the SSH attack surface
- Still allows remote access and management using existing SSH commands and tools
- Secures Linux endpoints on-premise and in the cloud
- Stops SSH key abuse and sprawl
- Audits, records, logs, and stores all sessions (live and recorded)

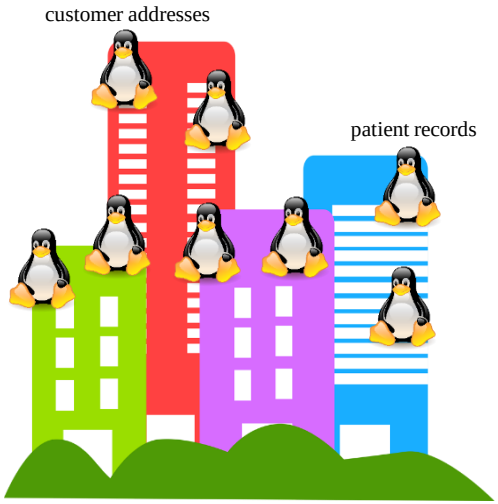
The Back Story

- Working on our AD Bridge product
- Placed Linux servers into the cloud for testing
- Within minutes, they were being scanned by hackers / bots
- Reviewed our Linux logs and saw all the username and password attempts

```
Nov 16 22:18:19 a-rh76-3 unix_chkpwd[18979]: password check failed for user (root)
Nov 16 22:18:23 a-rh76-3 unix_chkpwd[18983]: password check failed for user (root)
Nov 16 22:18:31 a-rh76-3 unix_chkpwd[18993]: password check failed for user (root)
Nov 16 22:18:32 a-rh76-3 sshd[18988]: HAPI: authentication failed. Error=The user
name or password is incorrect
Nov 16 22:18:34 a-rh76-3 unix_chkpwd[18998]: password check failed for user (test1)
Nov 16 22:18:42 a-rh76-3 unix_chkpwd[19013]: password check failed for user (root)
Nov 16 22:18:44 a-rh76-3 unix_chkpwd[19020]: password check failed for user (root)
Nov 16 22:18:44 a-rh76-3 sshd[19014]: reverse mapping checking getaddrinfo for
123.45.67.890 .asianet.co.in [123.45.67.890] failed - POSSIBLE BREAK-IN ATTEMPT!
Nov 16 22:18:45 a-rh76-3 unix_chkpwd[19024]: password check failed for user (root)
Nov 16 22:18:46 a-rh76-3 unix_chkpwd[19029]: password check failed for user (root)
Nov 16 22:18:50 a-rh76-3 unix_chkpwd[19034]: password check failed for user (root)
Nov 16 22:18:50 a-rh76-3 unix_chkpwd[19036]: password check failed for user (root)
Nov 16 22:18:52 a-rh76-3 unix_chkpwd[19040]: password check failed for user (root)
Nov 16 22:18:55 a-rh76-3 unix_chkpwd[19046]: password check failed for user (root)
Nov 16 22:19:00 a-rh76-3 unix_chkpwd[19051]: password check failed for user (root)
Nov 16 22:19:16 a-rh76-3 unix_chkpwd[19060]: password check failed for user (root)
Nov 16 22:19:42 a-rh76-3 sshd[19073]: reverse mapping checking getaddrinfo for
123.45.67.890.alfanet24.pl [123.45.67.890] failed - POSSIBLE BREAK-IN ATTEMPT!
Nov 16 22:19:43 a-rh76-3 unix_chkpwd[19080]: password check failed for user (root)
Nov 16 22:19:52 a-rh76-3 unix_chkpwd[19089]: password check failed for user (root)
Nov 16 22:20:09 a-rh76-3 unix_chkpwd[19115]: password check failed for user (root)
```

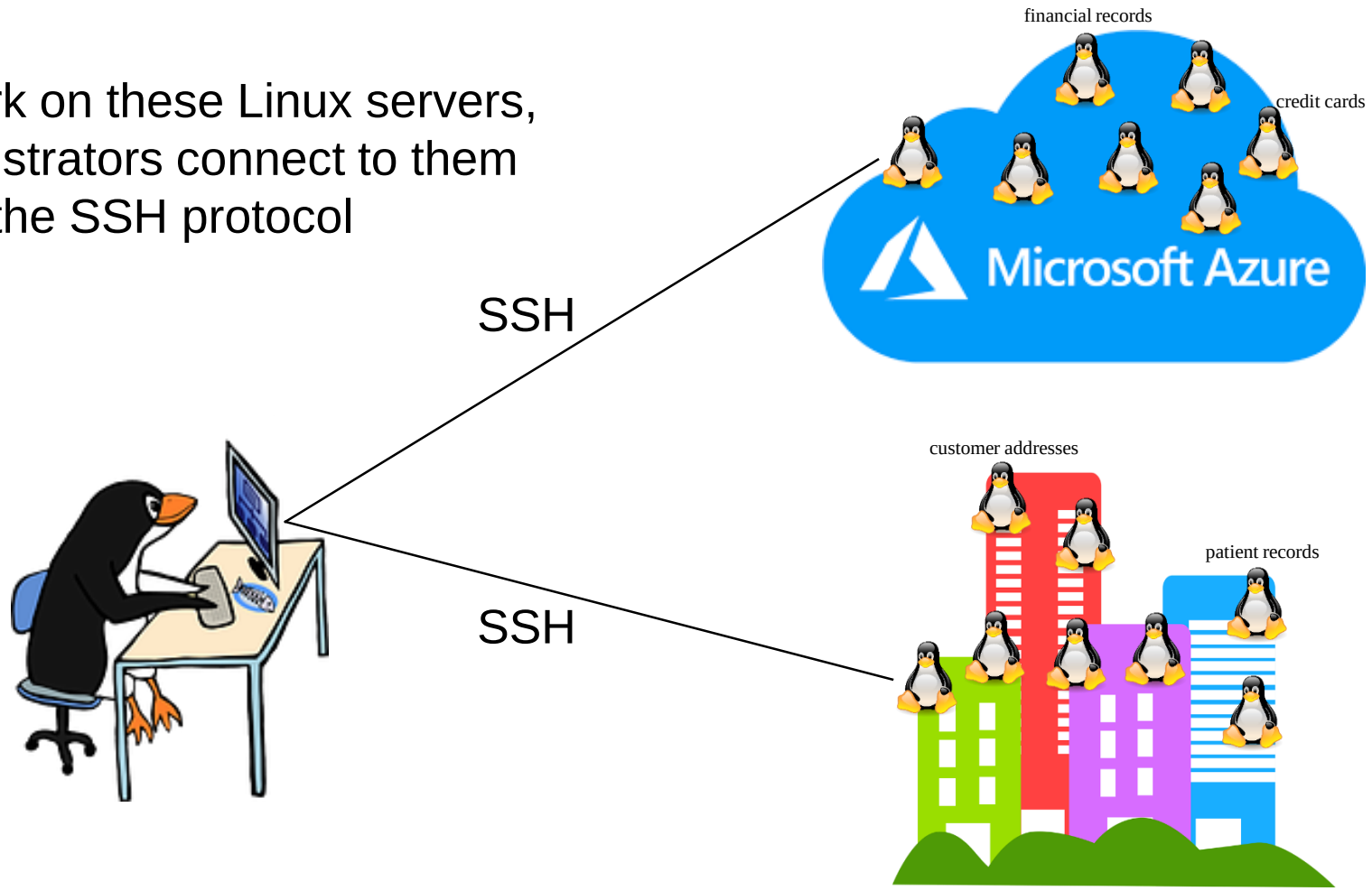
Millions of Linux endpoints are deployed across enterprises with more being deployed every day

These endpoints often contain confidential corporate and customer information, run critical apps, and perform essential functions for the business

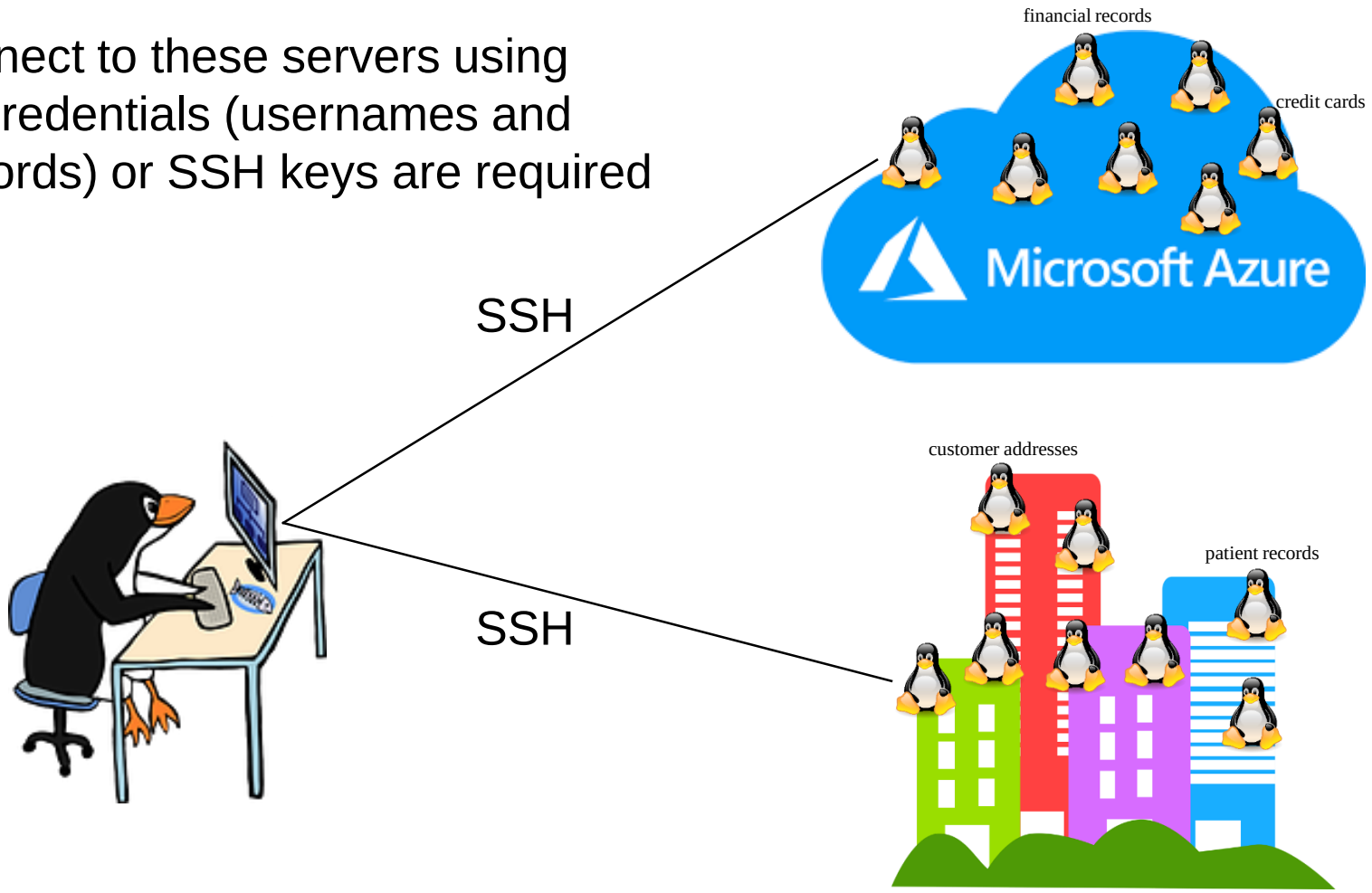


- Credit cards
- Patient data
- Customer records
- Financial statements
- Legal documents
- SAP
- Oracle

To work on these Linux servers, administrators connect to them using the SSH protocol



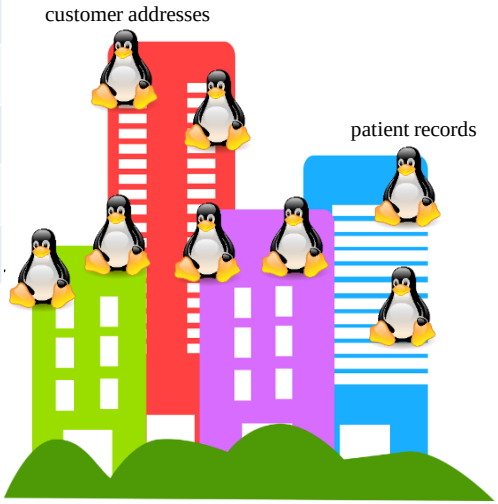
To connect to these servers using SSH, credentials (usernames and passwords) or SSH keys are required



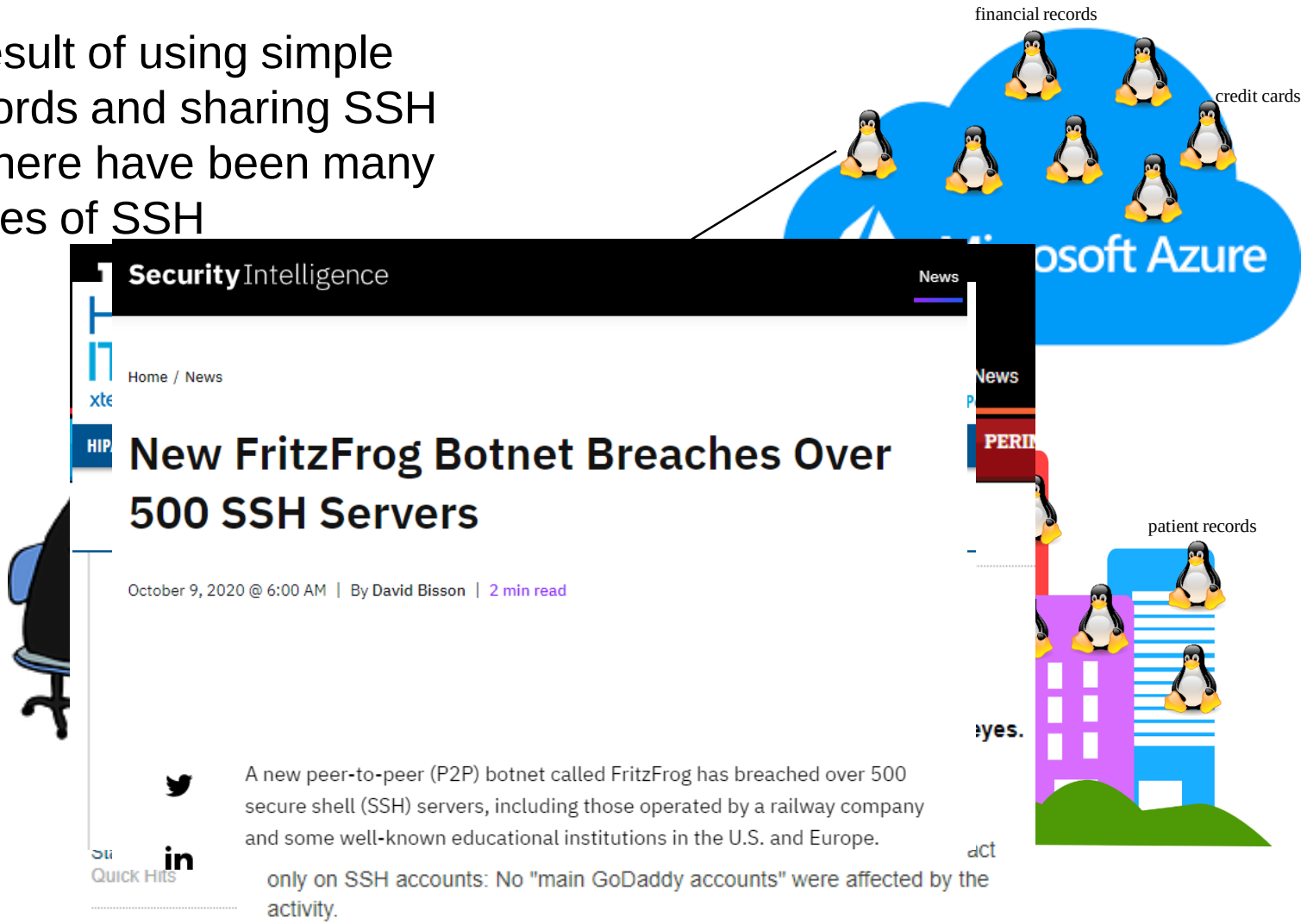
To make it easy to remember, administrators often use simple usernames and passwords

Top Usernames	Top Passwords
root	1234
admin	admin
user	password
test	default
guest	admin1234
ftpuser	ftp
support	Test

Source: F5 Labs

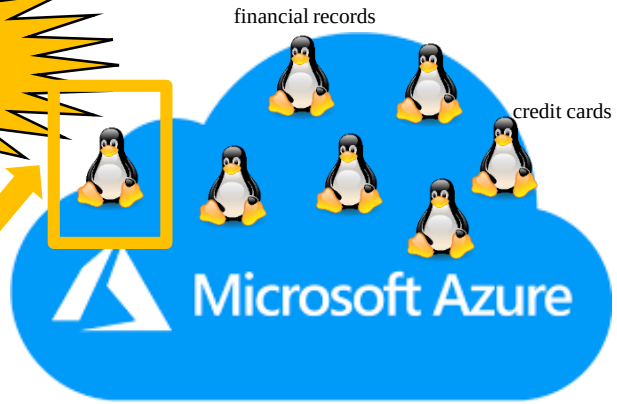


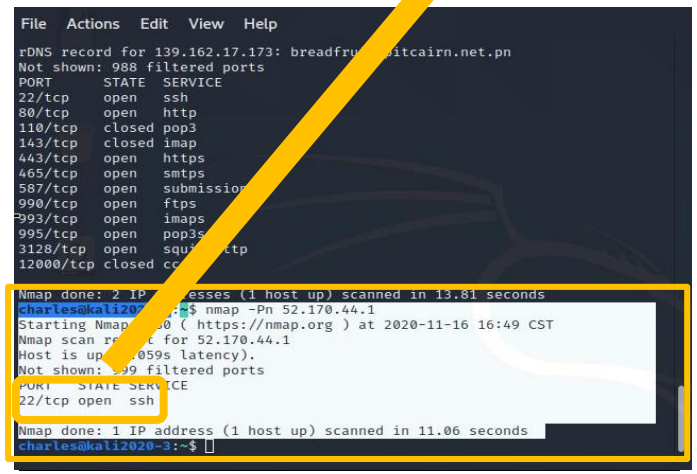
As a result of using simple passwords and sharing SSH keys, there have been many breaches of SSH

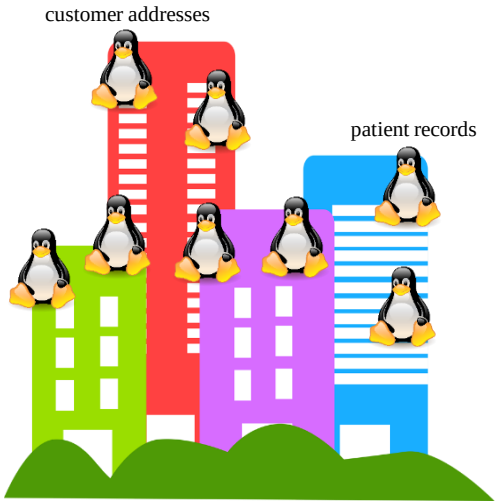


Hackers know that these have valuable information constantly scanning these Linux endpoints looking for open SSH ports

Open SSH Port discovered!!

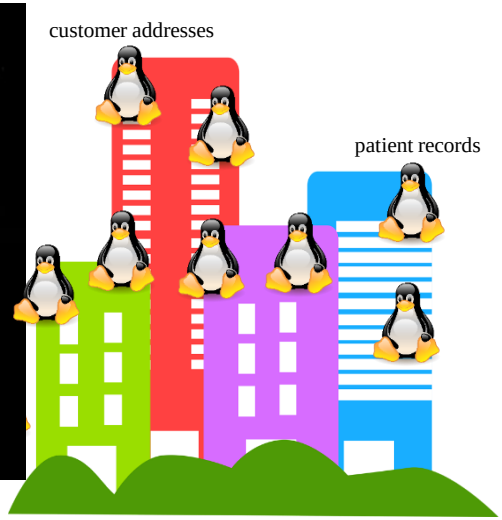






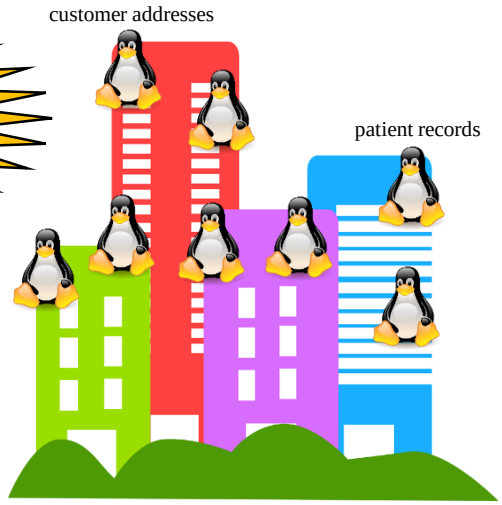
To see how hackers have attempted to penetrate your Linux servers, take a look at your Linux logon ‘secure’ system log where every logon attempt is captured.

Once the hacker finds an open SSH port, he/she can use a Dictionary of terms to attempt to access the Linux endpoint



Once the hacker has penetrated one endpoint, he/she now has horizontal access to other Linux endpoints to look for valuable data

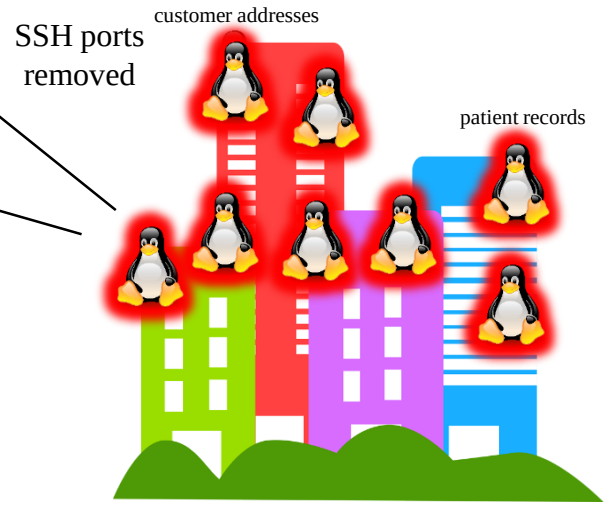
Credit Cards Compromised!!



*SSH*epherd makes Linux SSH Invisible & Secure

SSHepherd cloaks SSH so that the SSH attack surface is completely removed

SSH



- No sshd services are running and no open SSH firewall ports
- SSH ports cannot be scanned
- Continue using your Key Management solutions, SSH tools, and scripts – while preventing unauthorized individuals from even seeing the interfaces

Hackers will not see SSH

```
C:\Users\admin>nmap 192.168.1.34 -sS -sV -sC -v

Starting Nmap 7.40 ( https://nmap.org ) at 2018
```

No SSH ports will show up in the scan

Multi-Layered Security is a Best Practice

What are security administrators currently doing to help prevent SSH hacks?

- Using a Key Management system and stronger passwords
  - Yes, but the hackers have automated dictionary and other brute force techniques
- Port knocking
  - Helpful but still vulnerable to man in the middle attacks
- Restrict access to specific user IDs and ports
  - Useful but a hassle for the administrator
- Temporary rendezvous points
  - Good but can require a lot of configuration work/scripts and it's hard with automated workflows
- Lock down the shell
  - Helpful but configuration file can get overwritten, or updated where something got missed and discovered too late
- Disable Root login
  - Helpful but the hackers can still get in other ways (sudo)
- Change the default SSH ports
  - Yes, but the hackers will still find it if it can be scanned

What are critical features for a multi-layered secure SSH solution?

It needs to remove the SSH attack surface entirely,

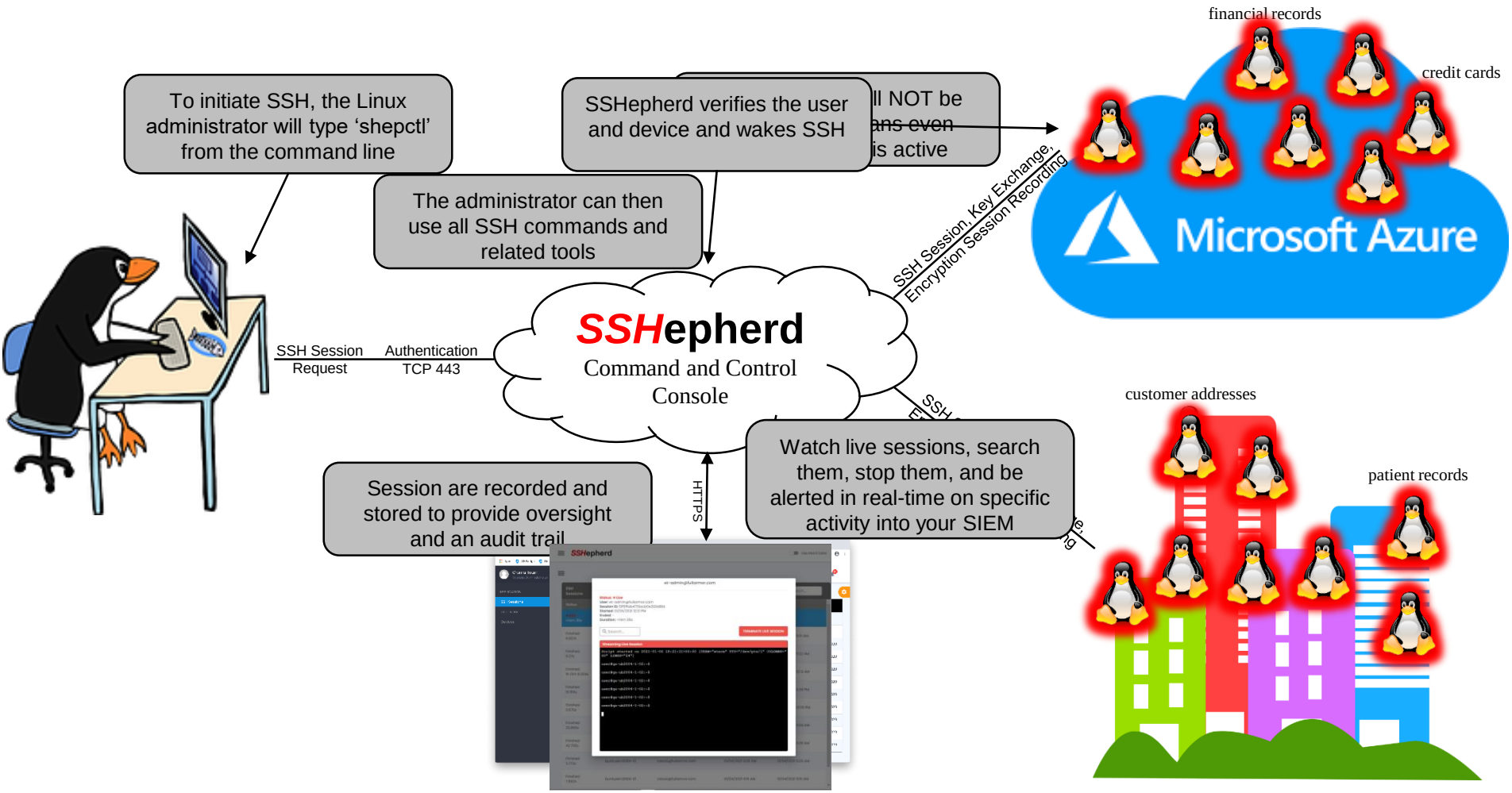
It must be easy for the Linux setup and administer

- ✓ Remove the SSH Attack Surface
- ✓ Work with existing SSH commands, Key Management solutions, and DevOps tools
- ✓ Easy setup
- ✓ Audit and track all activity, live and recorded

It must work with existing tools, scripts, and DevOps processes

It should have both a command line and web UI interface

SSHepherd Process



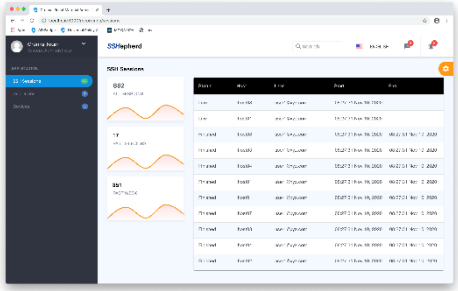
SSHepherd Setup

Setup Administrator

1) Install the SSHepherd console (either in your Azure tenant or on-premise)

2) Install SSHepherd agents to the Linux endpoints whether on-premise or in the cloud

3) Enter the user and device combinations to create SSH access groups



Linux SSH User

1) Install the SSHepherd cmd line tool

Apt/yum/zypper install shepctl

2) Type ‘shepctl’ to initiate a secure SSHepherd tunnel

Shepctl –h for Help and detailed usage instructions

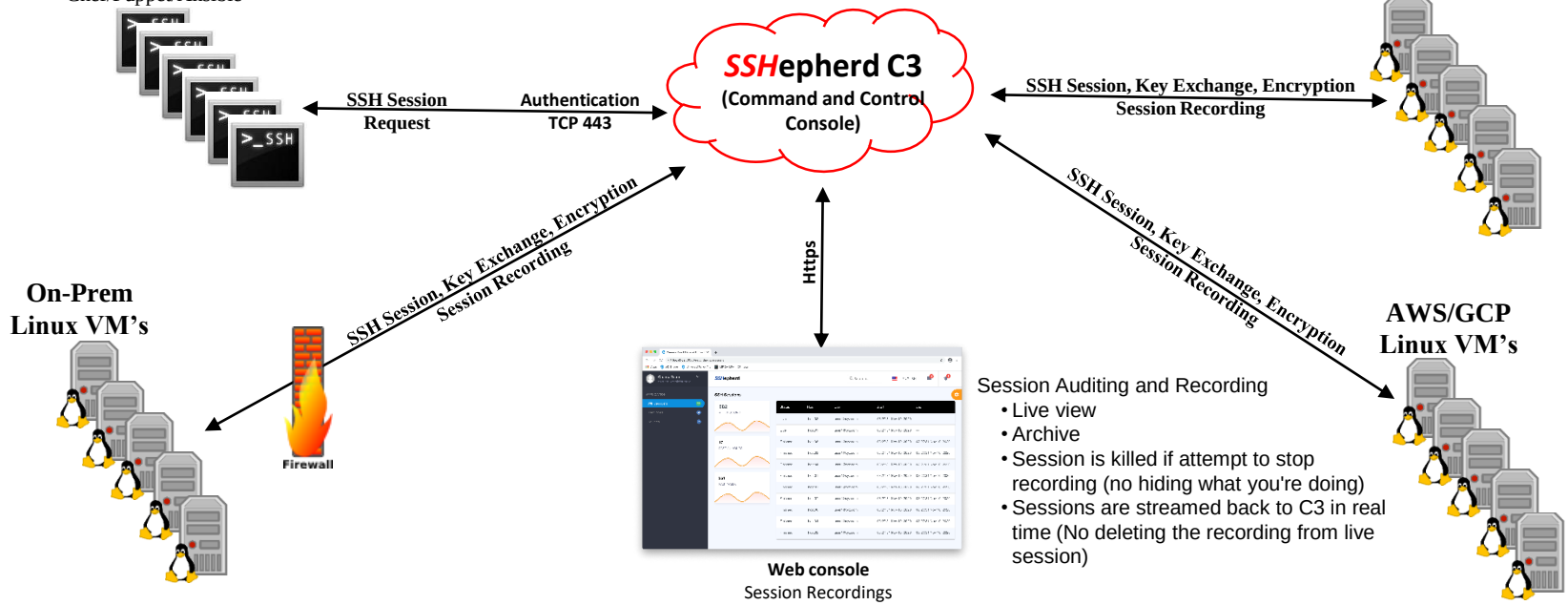
3) Continue using SSH, Key management, scripts, and automation tools

puppet CHEF

Red Hat Ansible Automation Platform

SSHepherd Architecture

- Multi-Factor Authentication
- Authorization (Access Management)
- Session Recording (Live and Archive View)



Full Armor Contact

Rich Farrell

rfarrell@fullarmor.com

SSHepherd

Removing the SSH Attack Surface