Cmd Whitepaper:

The Essential Eight: Cmd Mapping _

cmd.com

The Essential Eight

When it comes to Linux environments, we’re always looking for best practices guides, implementation strategies and hardening frameworks that allow us to better inform our customers on how to get ahead of adversarial behaviour effectively.

In the past we’ve mapped to PCI, ISO and even SOC2 standards through mapping, but a growing market demand in Australia has tasked us with mapping a number of Cmd’s enterprise capabilities to the Essential Eight mitigation strategies released by the ACSC.

The ACSC built out the Essential Eight mapping with a fairly targeted goal, well articulated by the summary on their aforementioned page :

> *“While no single mitigation strategy is guaranteed to prevent cyber security incidents, organisations are recommended to implement eight essential mitigation strategies as a baseline.* ***This baseline, known as the Essential Eight, makes it much harder for adversaries to compromise systems****. Furthermore, implementing the Essential Eight proactively can be more cost-effective in terms of time, money and effort than having to respond to a large-scale cyber security incident.”*

There’s a few key components to mapping a functionality of the Essential Eight guidance to your organization, and thankfully a number of helpful guides have been created to focus security engineering efforts in a way that can get ahead of some of the guesswork. As a starting point, we used the Linux Environment guide as a core to our research, mapping some of the capabilities of our enterprise product.

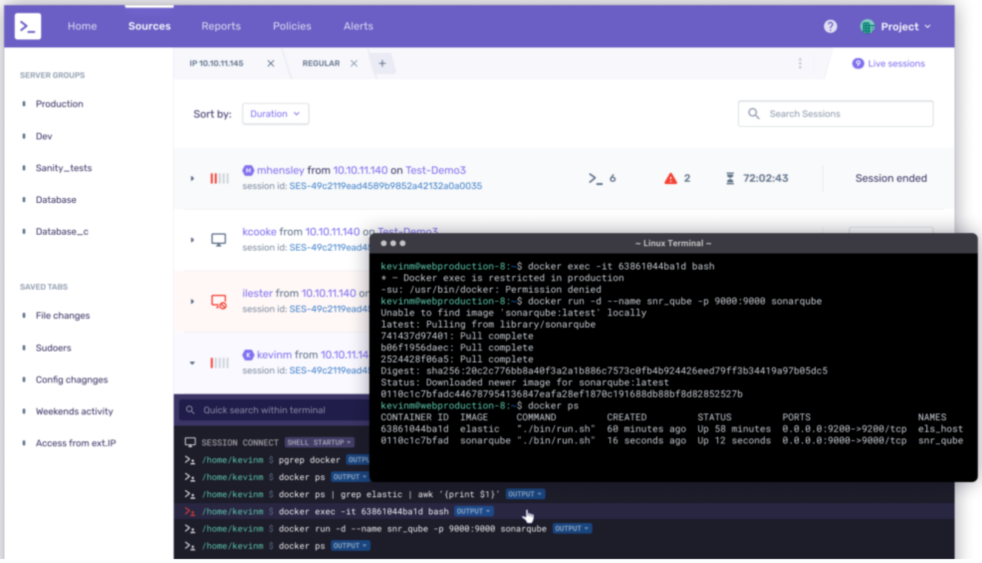
We’ll be covering how Cmd can assist in adhering to a number of the criteria outlined in the Linux Infrastructure below, and continue with a formal guide in assessing your maturity level in a second whitepaper leveraging Cmd’s reporting capabilities.

Application control

Application control is a speciality of Cmd, specifically through our Trigger engine that can either target File interaction or Binary Invocation. You’ve got a few standard profiles that can specifically allow-list, or deny-list a series of binaries available on the system, including third-party applications that are installed as part of a non-native linux distribution.

Cmd leverages a series of mechanisms to allow you to constrain application execution within your environment, with escalating levels of response; For example, rather than blocking access to Hashicorp Vault via the CLI, you could invoke a Multi-factor authentication prompt to enforce the control.

Another example of Application control can be implemented by using File Triggers within the Cmd interface to restrict changes to configurations, binaries and/or directories that may perform sensitive actions, or be prone to tampering. An implementation we have seen to protect standard system binary directories is to restrict file write operations within known, sensitive locations.



mhensley from 10.10.11.140 on Test-Demo3
session id: SES-49c2119ead4589b9852a42132a0a0035
>_ 6 ⚠ 2 ⧗ 72:02:43 Session ended

kcooke from 10.10.11.140 on Test-Demo3
session id: SES-49c2119ead4

ilester from 10.10.11.140 or
session id: SES-49c2119ead4

kevinm from 10.10.11.14
session id: SES-49c2119ead4

```
kevinm@webproduction-8:~$ docker exec -it 63861044ba1d bash
* - Docker exec is restricted in production
-su: /usr/bin/docker: Permission denied
kevinm@webproduction-8:~$ docker run -d --name snr_qube -p 9000:9000 sonarqube
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
741437d97401: Pull complete
b06f1956daec: Pull complete
2524428f06a5: Pull complete
Digest: sha256:20c2c776bb8a40f3a2a1b886c7573c0fb4b924426eed79ff3b34419a97b05dc5
Status: Downloaded newer image for sonarqube:latest
0110c1c7bfadc446787954136847eafa28ef1870c191688db88bf8d82852527b
kevinm@webproduction-8:~$ docker ps
CONTAINER ID  IMAGE     COMMAND        CREATED         STATUS        PORTS                   NAMES
63861044ba1d  elastic   "./bin/run.sh"  60 minutes ago  Up 58 minutes 0.0.0.0:9200->9200/tcp  els_host
0110c1c7bfad  sonarqube "./bin/run.sh"  16 seconds ago  Up 12 seconds 0.0.0.0:9000->9000/tcp  snr_qube
```

```
SESSION CONNECT  SHELL STARTUP
>_ /home/kevinm $ pgrep docker  OUTPU
>_ /home/kevinm $ docker ps  OUTPUT
>_ /home/kevinm $ docker ps | grep elastic | awk '{print $1}'  OUTPUT
>_ /home/kevinm $ docker exec -it 63861044ba1d bash  OUTPUT
>_ /home/kevinm $ docker run -d --name snr_qube -p 9000:9000 sonarqube  OUTPUT
>_ /home/kevinm $ docker ps  OUTPUT
```

Application and operating system patching

As many administrators that leverage Cmd today operate with a “Cattle, not Pets” mentality to their infrastructure, we’ve not deeply integrated capabilities to add patching and compliance to your fleet within our core capabilities. It is however, important to note that Cmd can be leveraged to validate that actions have been performed on a specific system through Administrative auditing and our CQL query language.

A common use-case for our CQL query language is for ensuring that operational tasks are being completed in an audit-friendly way. A quick CQL search for specific binaries such as apt, rpm and/or dpkg can provide some insights as to how frequently invocations to manually update system settings and preferences.

As a side note, a capability of reporting system OS version and specific Kernel versions is included in our `details` view within any session to give you a quick overview of exactly what version of a system is deployed in your environment.

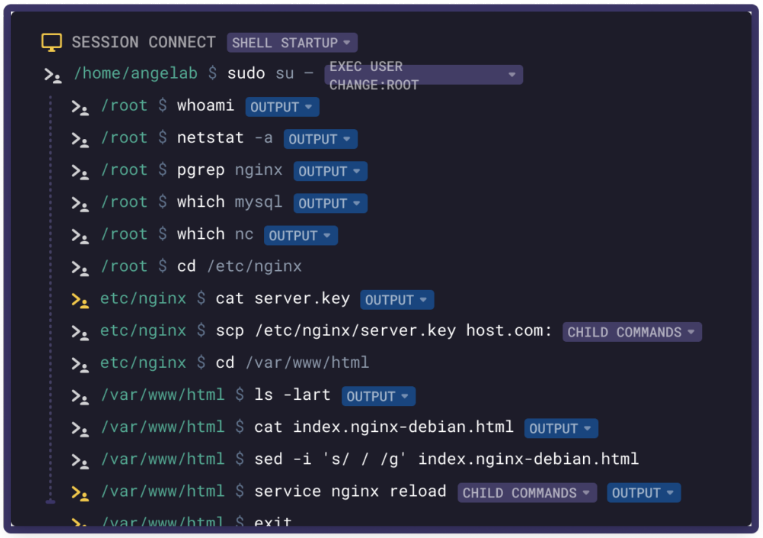
Configure Microsoft Office macro settings

Given we’ve never really seen Microsoft Word running within production infrastructure, we’d hazard a guess you haven't either - Check out some of the recommended hardening standards for Windows if you’re running windows apps within Wine on your infrastructure.

User application hardening

We’ve already covered some simple mapping and alignment within the Essential Eight for Application Control, and suggesting safe application hardening processes, including warning and/or blocking specific executions of binaries as privileged users (for example netcat operating as root). To get you started, Cmd has developed a series of standard best-practice implementation guides for known TTPs and sensitive actions running within your environment.

Specific File trigger actions may also be incorporated to ensure that a secure, hardened configuration is never changed by an end user for a “quick fix” that ends up compromising further services. Cmd enables you to defend against the common edge case and require any change is performed through standard sanctioned practices.



```
SESSION CONNECT
/home/angelab $ sudo su -
    /root $ whoami
    /root $ netstat -a
    /root $ pgrep nginx
    /root $ which mysql
    /root $ which nc
    /root $ cd /etc/nginx
    etc/nginx $ cat server.key
    etc/nginx $ scp /etc/nginx/server.key host.com:
    etc/nginx $ cd /var/www/html
    /var/www/html $ ls -lart
    /var/www/html $ cat index.nginx-debian.html
    /var/www/html $ sed -i 's/ / /g' index.nginx-debian.html
    /var/www/html $ service nginx reload
    /var/www/html $ exit
```

Restricting administrative privileges

Everyone needs root access, right? In many cases, infrastructure management requires heightened levels of privileged access, and often unfettered access during incident response or outage situations. Cmd provides a capability to restrict access to privileged accounts.

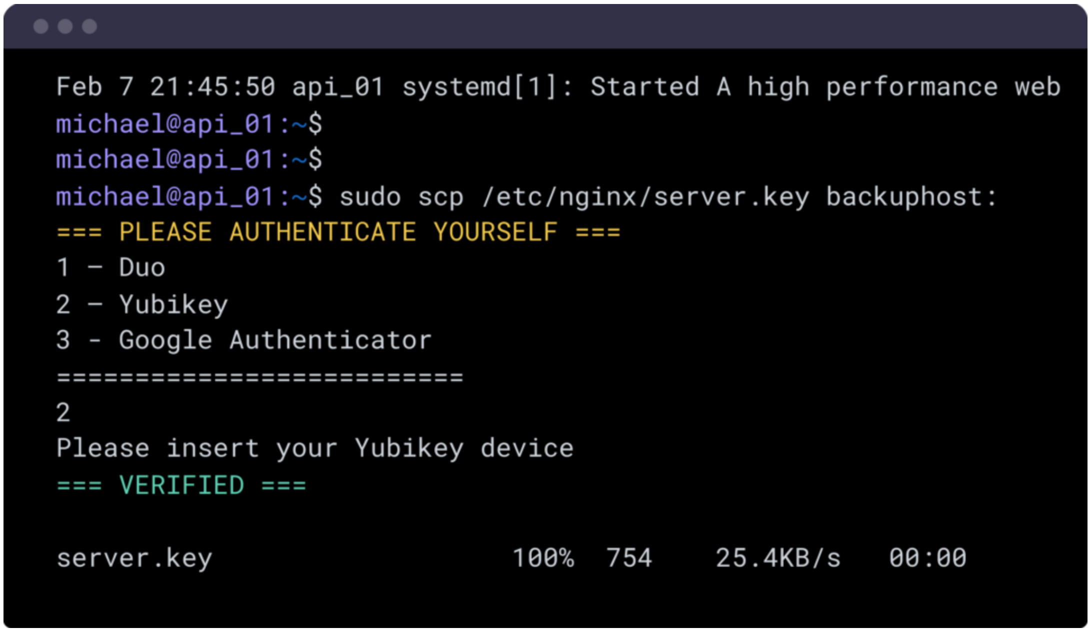
Given that much of the granted access to Linux systems is often granted in a gradual, drawn out process, often customers leverage Cmd to gain an understanding and baseline the access that many of their team members have today. Once identified, Cmd can assist in locking out access based on time of day, geolocation, risk level and more, alongside other standard account mitigation strategies such as removal of access or unused, legacy accounts.

Multi-factor authentication

A strong, diverse Multi-factor authentication strategy is critical to validate credentials for a user leveraging a hardware-tokens or push-based technology that is common with many providers such as Duo and Yubico.

Given that access must be granted to a small, specific group of users, a strong authentication practice is critical to ensuring that any access is as robust as possible. A key consideration security practitioners must consider while rolling out a solution is to provide capabilities to allow service based access, and constrain human interaction.

Cmd provides many mechanisms to perform these validation steps at the execution level, Session Invocation method and/or a stateful privilege escalation tool such as Sudo:

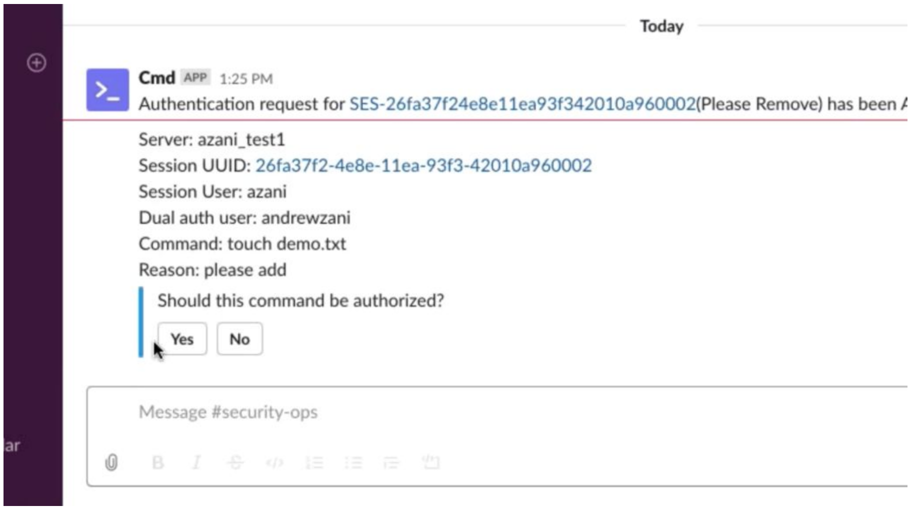


```
Feb 7 21:45:50 api_01 systemd[1]: Started A high performance web
michael@api_01:~$
michael@api_01:~$
michael@api_01:~$ sudo scp /etc/nginx/server.key backuphost:
=== PLEASE AUTHENTICATE YOURSELF ===
1 – Duo
2 – Yubikey
3 - Google Authenticator
==========================
2
Please insert your Yubikey device
=== VERIFIED ===

server.key                   100%  754    25.4KB/s   00:00
```

Daily backups

Like upgrading / updating procedures on Linux systems, Cmd does not specifically implement a backup methodology within the platform as many system tools offer these capabilities that may be customized to a specific environments use-case. It is critical to ensure that any backup process is monitored - as such, many customers implement Cmd alerting and webhook integrations to automatically report on processes (such as backup, service actions, or shut-downs) to central alerting services such as Slack:



Authentication request for SES-26fa37f24e8e11ea93f342010a960002(Please Remove) has been A

Server: azani_test1
Session UUID: 26fa37f2-4e8e-11ea-93f3-42010a960002
Session User: azani
Dual auth user: andrewzani
Command: touch demo.txt
Reason: please add

Should this command be authorized?

General hardening of Linux

While there's many hardening standards, guides and practices, those included with the Essential Eight are sufficient for many environments. This may be extended to include different TTP's that are detected with standard signatures and policies within Cmd.

Check out a number of the standard policy frameworks on our website for further information around alignment and details.