

Aus Cyber-Unfällen gelernt

Über die Lehren der Incident Response und die Widersprüche zwischen Prävention und akutem Vorfall

Dresdner Abwassertagung 2026

Jan Rähm



Agenda

1. Der Major Cyber Incident

2. Organisatorischer Ausnahmezustand

3. Menschlicher Ausnahmezustand

4. Technischer Ausnahmezustand

5. Fazit

1. Der Major Cyber Incident



2. Organisatorischer Ausnahmezustand



VERHALTEN BEI IT-NOTFÄLLEN



Ruhe bewahren & IT-Notfall melden
Lieber einmal mehr als einmal zu wenig anrufen!



IT-Notfallrufnummer:



Wer meldet?



Welches IT-System ist betroffen?



Wie haben Sie mit dem IT-System gearbeitet?
Was haben Sie beobachtet?



Wann ist das Ereignis eingetreten?



Wo befindet sich das betroffene IT-System?
(Gebäude, Raum, Arbeitsplatz)

Verhaltenshinweise

Weitere Arbeit
am IT-System
einstellen

Beobachtungen
dokumentieren

Maßnahmen nur
nach Anweisung
einleiten

Herausgeber: Bundesamt für Sicherheit in der Informationstechnik

Bild: www.bsi.bund.de

auf dem Server
verschlüsselt

Meldekettens mit
veralteten
Nummern

nie unter Stress
getestet

IT-Abteilung informiert alle
per Mail:
„Wir wurden gehackt. Wirklich“

Pressemitteilung sagt
„vorsorglich abgeschaltet“

gar nichts gesagt



3. Menschlicher Ausnahmezustand





Leugnung

Ärger

Feilschen

Depression


Akzeptanz



Eine IT-Krise ist kein Sprint,
Eine IT-Krise ist ein Marathon!

3. Technischer Ausnahmezustand





Wir haben das
Internet abgeschaltet!

Unser IT-Dienstleister
monitort uns über einen
dauerhaften VPN-Tunnel.

Natürlich segmentieren
wir unsere Netze!

Unser Admin,
der weiß das.

Wir haben
Backups!

```
2026-04-13 20:39:53,767 - helpers.py[DEBUG]: config-seed already ran (freq=once-per-instance)
2026-04-13 20:39:53,767 - handlers.py[DEBUG]: finish: init-local/config-seed_random: SUCCESS: config-seed_random previously ran
2026-04-13 20:39:53,767 - modules.py[DEBUG]: Running module mounts (<module 'cloudinit.config.cc_mounts' from '/usr/lib/python3/dist-packages/cloudinit/config/cc_mounts.py'>) with frequency once-per-instance
2026-04-13 20:39:53,767 - handlers.py[DEBUG]: start: init-local/config-mounts: running config-mounts with frequency once-per-instance
2026-04-13 20:39:53,767 - helpers.py[DEBUG]: config-mounts already ran (freq=once-per-instance)
2026-04-13 20:39:53,767 - handlers.py[DEBUG]: finish: init-local/config-mounts: SUCCESS: config-mounts previously ran
2026-04-13 20:39:53,767 - modules.py[DEBUG]: Running module set_hostname (<module 'cloudinit.config.cc_set_hostname' from '/usr/lib/python3/dist-packages/cloudinit/config/cc_set_hostname.py'>) with frequency once-per-instance
2026-04-13 20:39:53,767 - handlers.py[DEBUG]: start: init-local/config-set_hostname: running config-set_hostname with frequency once-per-instance
2026-04-13 20:39:53,767 - helpers.py[DEBUG]: config-set_hostname already ran (freq=once-per-instance)
2026-04-13 20:39:53,767 - handlers.py[DEBUG]: finish: init-local/config-set_hostname: SUCCESS: config-set_hostname previously ran
2026-04-13 20:39:53,767 - modules.py[DEBUG]: Running module cc_update_hostname (<module 'cloudinit.config.cc_update_hostname' from '/usr/lib/python3/dist-packages/cloudinit/config/cc_update_hostname.py'>) with frequency always
2026-04-13 20:39:53,768 - handlers.py[DEBUG]: start: init-local/config-cc_update_hostname: running config-cc_update_hostname with frequency always
2026-04-13 20:39:53,768 - helpers.py[DEBUG]: Attempting to update hostname to moonraker in 0 files
2026-04-13 20:39:53,768 - distros[DEBUG]: Attempting to update hostname to moonraker in 0 files
2026-04-13 20:39:53,768 - handlers.py[DEBUG]: finish: init-local/config-update_hostname: SUCCESS: config-update_hostname previously ran
2026-04-13 20:39:53,768 - modules.py[DEBUG]: Running module update_etc_hosts (<module 'cloudinit.config.cc_update_etc_hosts' from '/usr/lib/python3/dist-packages/cloudinit/config/cc_update_etc_hosts.py'>) with frequency always
2026-04-13 20:39:53,768 - handlers.py[DEBUG]: start: init-local/config-update_etc_hosts: running config-update_etc_hosts with frequency always
2026-04-13 20:39:53,769 - helpers.py[DEBUG]: Running config-update_etc_hosts using lock (<cloudinit.helpers.DummyLock object at 0x7fff59cf0050>)
2026-04-13 20:39:53,769 - util.py[DEBUG]: Reading from /etc/hosts (quiet=False)
2026-04-13 20:39:53,769 - util.py[DEBUG]: Reading 545 bytes from /etc/hosts
2026-04-13 20:39:53,769 - util.py[DEBUG]: Reading from /etc/cloud/templates/hosts.debian.tpl (quiet=False)
2026-04-13 20:39:53,769 - util.py[DEBUG]: Reading 845 bytes from /etc/cloud/templates/hosts.debian.tpl
2026-04-13 20:39:53,769 - templater.py[DEBUG]: Rendering content of '/etc/cloud/templates/hosts.debian.tpl' using rendererer jinja
2026-04-13 20:39:53,773 - util.py[DEBUG]: Writing to /etc/hosts - wb: [644] 545 bytes
2026-04-13 20:39:53,774 - handlers.py[DEBUG]: finish: init-local/config-update_etc_hosts: SUCCESS: config-update_etc_hosts ran successfully and took 0.005 seconds
2026-04-13 20:39:53,774 - modules.py[DEBUG]: Running module users_groups (<module 'cloudinit.config.cc_users_groups' from '/usr/lib/python3/dist-packages/cloudinit/config/cc_users_groups.py'>) with frequency once-per-instance
2026-04-13 20:39:53,774 - handlers.py[DEBUG]: start: init-local/config-users_groups: running config-users_groups with frequency once-per-instance
2026-04-13 20:39:53,774 - helpers.py[DEBUG]: config-users_groups already ran (freq=once-per-instance)
2026-04-13 20:39:53,774 - handlers.py[DEBUG]: finish: init-local/config-users_groups: SUCCESS: config-users_groups previously ran
2026-04-13 20:39:53,774 - modules.py[DEBUG]: Running module ssh (<module 'cloudinit.config.cc_ssh' from '/usr/lib/python3/dist-packages/cloudinit/config/cc_ssh.py'>) with frequency once-per-instance
2026-04-13 20:39:53,774 - handlers.py[DEBUG]: start: init-local/config-ssh: running config-ssh with frequency once-per-instance
2026-04-13 20:39:53,774 - helpers.py[DEBUG]: config-ssh already ran (freq=once-per-instance)
2026-04-13 20:39:53,774 - handlers.py[DEBUG]: finish: init-local/config-ssh: SUCCESS: config-ssh previously ran
2026-04-13 20:39:53,774 - modules.py[DEBUG]: Running module set_passwords (<module 'cloudinit.config.cc_set_passwords' from '/usr/lib/python3/dist-packages/cloudinit/config/cc_set_passwords.py'>) with frequency once-per-instance
2026-04-13 20:39:53,774 - handlers.py[DEBUG]: start: init-local/config-set_passwords: running config-set_passwords with frequency once-per-instance
2026-04-13 20:39:53,775 - helpers.py[DEBUG]: config-set_passwords already ran (freq=once-per-instance)
2026-04-13 20:39:53,775 - handlers.py[DEBUG]: finish: init-local/config-set_passwords: SUCCESS: config-set_passwords previously ran
2026-04-13 20:39:53,775 - main.py[DEBUG]: Ran 8 modules with 0 failures
2026-04-13 20:39:53,775 - util.py[DEBUG]: Reading from /proc/uptime (quiet=False)
2026-04-13 20:39:53,775 - util.py[DEBUG]: Reading 11 bytes from /proc/uptime
2026-04-13 20:39:53,775 - atomic_helper.py[DEBUG]: Atomically writing to file /var/lib/cloud/dbat/status.json (via temporary file /var/lib/cloud/data/tmpw92kcebg) - w: [644] 567 bytes/chars
2026-04-13 20:39:53,775 - performance.py[DEBUG]: cloud-init stage: 'init-local' took 0.114 seconds
2026-04-13 20:39:53,775 - handlers.py[DEBUG]: finish: init-local: SUCCESS: searching for local datasources
2026-04-13 20:39:53,775 - socket.py[INFO]: Sending sd_notify(STATUS=Waiting on external services to complete before starting the network stage.)
2026-04-13 20:39:53,815 - performance.py[DEBUG]: Waiting to start stage network took 0.039 seconds
2026-04-13 20:39:53,815 - socket.py[INFO]: Sending sd_notify(STATUS=Running (network stage))
2026-04-13 20:39:53,815 - signal_handler.py[INFO]: Signal state <function 'handle_exit' at 0x7fff5b152200> - previously custom handler.
2026-04-13 20:39:53,815 - signal_handler.py[INFO]: Signal state <function 'handle_exit' at 0x7fff5b152200> - previously custom handler.
```

Unsere Firewall loggt ... für sieben Tage.

Unsere E3-Lizenz enthält Logging!

Der Log-Server liegt auf dem HyperV.

Bild: Jan Rahim, Hisolutions.com

4. Fazit





Es ist keine Frage, OB es passiert – sondern WANN – und ob Sie dann vorbereitet sind.

Der schlimmste Zeitpunkt, Notfallplan zum ersten Mal zu lesen, ist während des Major Cyber Incidents.

Wasseraufbereitungs- und -behandlungsanlagen hören nicht auf zu arbeiten, weil die IT steht. Sie zu weiter steuern zu können, ist die Herausforderung.

2022 – Fallbeispiel ASG (Abwasser- u.
Straßenreinigungsbetrieb der Stadt Gifhorn)

<https://www.youtube.com/watch?v=OazNTpO9J9I>



Schloßstraße 1 | 12163 Berlin

info@hisolutions.com | +49 30 533 289 0

www.hisolutions.com