
Lubomir Bogda
Hronska 13
Kosice, 04011

>> recvmsgs@gmail.com https://github.com/recvmsgs <<

          0949-646-314

==========

>
Languages:

  English   -- C1-C2 (understanding)
                  C1-C2 (speaking)

  German   -- B2   (understanding)
                  A2-B1 (speaking)

==========

    ``````````````````
    ` 2021, 4 months
    ` Numerika O.Z.
    ` (volunteer work)
    `
    ``````````````````

>
ERPNext   -- Building, backing up, installation, migration of ERPNext and its component Frappe
                  framework. Linux, Apache, MariaDB, Python, virtualenv, Honcho, SystemD, Git etc.
                  This was quite challenging because there was no (or obsolete) documentation.
                  I had to modify Ansible files, change git flags, debug code, just to find out i.e.
                  how to build the most recent stable version or how to upgrade without losing data.

>
KVM hypervisor
               -- Some jobs were done on commandline (virsh, virt-*, libvirt) but most work was done
                  using virt-manager GUI.

               -- For reference, please contact Michal Mikulaško at:
                  info@numerika.sk or https://sk.linkedin.com/in/michal-mikulasko-914921182

    ````````````````
    ` 2019 - now
    ` Tiger Nets s.r.o.
    ` (on contract)
    `
    ````````````````

>
Chia farming:
               -- Chia cryptocurrency mining: a master node with SSD storage within Hyper-V plus almost
                  100 harvester nodes running within Docker containers on Hyper-V which were connected to
                  Google drive mounts using Rclone. Deployment and changes were done using Bash scripts;
                  I also wrote SystemD Unit files. Before I could automate it all using Ansible, Google
                  introduced two-factor authorisation so the project was cancelled.

>
Windows Server 2019:

               -- Easy job except for necessary downtimes during upgrades. Mostly used to run Hyper-V
                  or Stormware POHODA. Active Directory wasn't needed.
>
Hyper-V:
               -- Cloning, checkpointing, snapshoting, some use of templates, virtual LAN.
>
VMware vSphere:

               -- Version 4.1 which was released in 2011, therefore my knowledge doesn't reflect current
                  versions. Same tasks done as with Hyper-V (see above).

>
LibreNMS:
-- Setting up Regex- or SQL- based alert rules. I helped with maybe 150 hosts' monitoring.

-- For reference, please contact Peter Pieger at:
office@tigernets.sk or +421-905-604-392

```
2019-now
BARNET s.r.o.
(on contract)
```

>
BIND, Docker,
Gitlab: -- Deployed a backup DNS server using Git, Gitlab and Dockerfiles. This was just 'git clone/add/commit' after editing just zone file's IP addessr; then add 'docker run' to SystemD unit file.

>
LAMP stack: -- Simple shared hosting with Linux, Apache, MySQL, PHP; DNS, FTP and e-mail servers. Concerning my duties, it was essentially like working for e-SOLUTIONS (scroll below).

-- For reference, please contact Jan Barger at:
https://sk.linkedin.com/in/jan-barger-67172062

```
2018
Treecom s.r.o.
(on contract)
```

>
Docker, Nginx,
node.js, Asterisk:

-- Helped deploy several web servers running Nginx and node.js + Redis / PostgreSQL in Docker containers using Docker Compose and Docker Swarm. Although I understand YAML syntax of Compose files, I mostly used files from Hub or Github / Gitlab with minor changes. Docker was running within RancherOS (which is no longer mainteined upstream).

-- Asterisk testing with Docker:
A simple two-node network of Docker containers to automate testing of Asterisk: writing Dockerfiles, simple Docker Compose use, deployment using Docker Machine on Debian host with Containers running Alpine Linux, BASH scripts to test VOIP (SIP) connectivity.

-- For reference, please contact Martin Bucko i.e. on
https://sk.linkedin.com/in/bucko

```
2016-2017
Napri s.r.o.
(on contract)
```

>
LAMP stack -- Simple shared hosting with Linux, Apache, MySQL, PHP; DNS, FTP and e-mail servers. Concerning my duties, it was essentially like working for e-SOLUTIONS (scroll below). The difference was that now I ran the systems within KVM hypervisor.

-- For reference, please contact Jan Barger at:
https://www.napri.sk/

```
`````````````````
` 2015
` SAP.com Bratislava
`
`````````````````
```

>
Docker, Jira

-- Docker container administration, monitoring and root cause analysis:
   It was a complex DevOps pipeline built with Saltstack, Docker, GIT, Jenkins, Jira, ELK;
   running in an AWS cloud. Note: My experience was limited to administering the Ubuntu
   host OS and its containers. I did not work with Saltstack or the infrastructure
   (storage and networking). I didn't participate in Dev part.
   My experience with their CD/CD tools was limited.

-- For reference, please contact Juraj Kniz i.e. on
   https://sk.linkedin.com/in/juraj-kniz-1865318

---

```
`````````````
` 2014 - 2015:
` freelance work
`
`````````````
```

>
(sample tasks):

-- Linux RAID (dmraid) monitoring setup; softraid and LSI raid status monitoring:
   Cacti, Nagios NRPE installation; softraid storage; design optimalization:
   Decided to send state information from mdadm directly to syslog server avoiding overhead
   and unnecessary latency by implementing status polling, daemon mode and syslog().

-- Darknet to Internet Mail gateway: Helped with deployment, scripting, testing and
   troubleshooting on a simple NATted cluster (no HA or load balancing) on VMware vSphere.

---

```
`````````````````
` 2012 - 2014
` eSOLUTIONS s.r.o.
`
`````````````````
```

>
Debian GNU/Linux:

(practice): -- Lots of troubleshooting, incidents, root cause analyses, using diagnostic tools like
   strace, lsof routinely, even gdb and kdb sometimes.
   I love to solve problems, getting into flow state, diving into the command line.

   With enough time I love to make a thorough root cause analysis, even for free.

(know-how): -- I  base my work on the ablility to learn required in-depth knowledge in shortest time
   possible, using any sources, learning jargon and concepts. I found out many times my
   'original' idea is in fact a part of a common practice with its own buzzword.

   I study concepts of the subject of my work before even starting to commit.
   I will gladly study documentation like UNIX manpages - sections {2,5,4,8,7}, Open Group
   POSIX, IETF RFCs. I have gathered some deeper knowledge of APT, dpkg, debconf.
   I understand Filesystem Hierarchy Standard, therefore I know where in the filesystem
   specific components, systems, subsystems or packages belong.
   (For example, I understand reasoning behind /usr merge that Freedesktop did).

   I underestand Debian project tools: debianutils, dschroot, debootstrap, devscripts, etc.
   I prefer to study source code related to software I work with, even in my spare time.

```
(know-how):     -- I have developed a special workflow, quickly filtering output, generating reports out of
                   more or less structured information coming from my local system and the Internet and
                   combining it together. It is crucial to me to understand the system thoroughly.

                   I don't rely on intuition when committing - before running a command I haven't run for
                   a long time I filter out the relevant information from a manual page.

                   I am careful when the result of my action is dependent on multiple factors:
                   I do dry-runs on tmpfs, turn on the no-op switch, edit a copy of a config file first,
                   genereate a list of separate commands and pass them to the interpreter over pipe after
                   checking them for correctness.
                   I don't run commands blindly without testing first. I believe too much confidence is
                   dangerous if your job is being a sysadmin.


>
Shell:
  bash,
  busybox sh,
  mksh,
  zsh           -- I have an eye for spotting compatibility problems like bashisms or bad quoting.

                   I commonly use 2-3 'one-liners' 2 or 3 lines long (with 174 characters per line).

                   I have an internal substitute for shell grammar changing Bournegol into something
                   more readable and comprehensible.
                   If I am not the only one to read it, I have a readline hack for it: Define my code as
                   a function in interactive shell session, expand aliases and save the compatible code.
                   Others mostly have no idea until I tell them or they see my workflow.

>
Regexp:
  extended      -- I use grep command with ERE to highlight particular text or characters in the console.
                   I wish to learn PCRE, if needed.

>
Editors:
  sed           -- I prefer a stream editor to have text file editing saved into shell history;
                   most of commands besides s/./&/g have been forgotten by now.
  AWK           -- Little consistent practice, pipelines using coreutils tools have been enough.

>
LAMP:
  Apache        -- Virtual hosts, mod_rewrite, htaccess.
  PHP           -- CGI,  FastCGI.
  MySQL         -- Common administration tasks, no HA, master-slave replication, binary log, schema rework.
  PostgreSQL    -- vanilla Debian setup, simple changes, some diagnostics.

>
Monitoring:
  Zabbix        -- I used a virtual appliance, 1.8 version. Very little practice with setup.

>
SMTP:
  Postfix       -- Virtual mailbox, home maildirs on one server.

>
POP3 & IMAP:
  Dovecot       -- Builtin SASL auth; Dovecot 1.x only.

>
DNS:
  BIND          -- 3rd level domain resolver for virtual hosting setup but not much more.

>
FTP:
  pure-ftpd     -- Virtual users in MySQL DB.

>
IPS, IDS, FW:
  fail2ban
  netfilter     -- No practice with rule matching, target extensions, masquerade target.
                   Using rule numbers with iptables is the top of my skills for now.


                -- For reference, please contact Ladislav Kucko at:
                   https://www.esolutions.sk/en/contact/
```

---

```
    ``````````````````````````
    ` 2013, 2+ months of july:
    ` O.Z. SOSNA (NGO)
    `
    ``````````````````````````
```

>
LAN,
Low end NAS,
Windows 7, w/o domain:

(abstract): -- Securing a LAN without a domain controller or Group Policy, a NAS as a SMB fileserver.
I learned to center my design around the user (they gave me a lot of feedback).

Registry configuration replication, Winlogon DLL pacthing, modified DACs,
automatic logon, priviledge separation, run as usability (...), UAC;
wake on LAN + suspend for a print server with frontend on end user workstations,
maintenance automatization:

Guest users had s separate account assigned, preventing them from running most SW.
I migrated infrequently accessed sensitive data to a server to centralize administration
and bypassed UAC prompt for priviledged users.
I made an attempt to make a simple AV scanner with a good detection engine by ESET and
SOPHOS work more like a full time antivirus with a real-time protection.
The workstation used as a print server was modified to go into ACPI sleep after a period
set from a remote interface besides stay-awake period and remote shutdown functionality.

The previous AV implementation's web shield had a bad detection rate.
I tried to overcome it by using a local (ht|f)tps? filtering AV proxy which ran a free
cmdline AV scanner with a good engine synchronously whenever a potentially exploitable
file format was downloaded, based on content-type header.
I tried to implement it using nativne Windows API, not a translation layer like Cygwin.

---

```
    ``````````````````````
    ` 2010, <3 months:
    ` T-Systems SK, Kosice
    `
    ``````````````````````
```

>
HP UNIX:

>
Shell:
HP-UX ksh -- My collegue with 2 years of experience at the company decided to run
'sed s/sh/ksh/g /etc/passwd > /etc/passwd'
over ssh in a for loop on several servers. I instructed him on the probable outcome.
File descriptors are allocated immediately so the file would be truncated.

>
Veritas VxFS -- I suspected that Oracle DB performance issue when backed by a file compared to a raw
block device could be solved by changing journal parameters on the mount.
My advice was to remount with noatime,data=writeback and the overhead was gone.

>
HP UNIX
(forgotten studies):
-- HA clustering, HP-UX LVM, LPARs, VxFS.

---

```
    ````````````````````````````
    ` until 2008:
    ` Gymnázium Tomáša Akvinského
    `
    ````````````````````````````
```

-- I actually learned all my skills, including languages and Linux by myself.
Anyway, I don't regret going to school, we were a great community.