

Healing a shellshocked infrastructure, in a couple of hours, from your bed

Michele Finelli
m@biodec.com
BioDec

Index

Shellshock

Fabric in few words

Know your infrastructure

Happy end !

Index

Shellshock

Fabric in few words

Know your infrastructure

Happy end !

Where were you the night between . . .

- ▶ The 24th of September, 2014, a vulnerability was announced.
- ▶ It was in a program called **bash**.
- ▶ It was **network exploitable**.
- ▶ It existed **since version 1.03 of Bash released in September 1989**.

Where were you the night between . . .

- ▶ The 24th of September, 2014, a vulnerability was announced.
- ▶ It was in a program called **bash**.
- ▶ It was **network exploitable**.
- ▶ It existed **since version 1.03 of Bash released in September 1989**.

Where were you the night between . . .

- ▶ The 24th of September, 2014, a vulnerability was announced.
- ▶ It was in a program called **bash**.
- ▶ It was **network exploitable**.
- ▶ It existed **since version 1.03 of Bash released in September 1989**.

Where were you the night between . . .

- ▶ The 24th of September, 2014, a vulnerability was announced.
- ▶ It was in a program called **bash**.
- ▶ It was **network exploitable**.
- ▶ It existed **since version 1.03 of Bash released in September 1989**.

Where were you the night between . . .

- ▶ The 24th of September, 2014, a vulnerability was announced.
- ▶ It was in a program called **bash**.
- ▶ It was **network exploitable**.
- ▶ It existed **since version 1.03 of Bash released in September 1989**.

No

Insert here your favourite funny picture of a cute cat showing amazement.

Patch time

Patch time

- Any Internet facing system that we managed had to be updated.
- We had to do that *fast*:

Patch time

- ▶ Any Internet facing system that we managed had to be updated.
- ▶ We had to do that *fast*:

Within an hour of the announcement of the Bash vulnerability, there were reports of machines being compromised by the bug. By 25 September 2014, botnets based on computers compromised with exploits based on the bug were being used by attackers for distributed denial-of-service (DDoS) attacks and vulnerability scanning.

Patch time

- ▶ Any Internet facing system that we managed had to be updated.
- ▶ We had to do that *fast*:

Within an hour of the announcement of the Bash vulnerability, there were reports of machines being compromised by the bug. By 25 September 2014, botnets based on computers compromised with exploits based on the bug were being used by attackers for distributed denial-of-service (DDoS) attacks and vulnerability scanning.

- ▶ Linux distribution maintainers did **an awesome job**.

Recap

- ▶ Hundreds of hosts.
- ▶ Fix them *now* !
- ▶ Do not make mistakes: after all the package is *bash*, if you botch it, it could be that the system will not allow login anymore ... or even that it will not reboot ...

Recap

- ▶ Hundreds of hosts.
- ▶ Fix them *now* !
- ▶ Do not make mistakes: after all the package is *bash*, if you botch it, it could that the system will not allow login anymore ... or even that it will not reboot ...

Recap

- ▶ Hundreds of hosts.
- ▶ Fix them *now* !
- ▶ Do not make mistakes: after all the package is *bash*, if you botch it, it could be that the system will not allow login anymore ... or even that it will not reboot ...

Recap

- ▶ Hundreds of hosts.
- ▶ Fix them *now* !
- ▶ Do not make mistakes: after all the package is *bash*, if you botch it, it could that the system will not allow login anymore ... or even that it will not reboot ...

Ah, just one more thing

Ah, just one more thing

- ▶ Insert here your favourite picture of Peter Falk playing “Columbo”.
- ▶ I was lying in my bed, because of a severe back-ache.

Index

Shellshock

Fabric in few words

Know your infrastructure

Happy end !

What is fabric

- ▶ Fabric is a Python (2.5-2.7) library and command-line tool for streamlining the use of SSH for application deployment or systems administration tasks.
- ▶ <http://www.fabfile.org/>
- ▶ Like a parallel shell, but better.

What is fabric

- ▶ Fabric is a Python (2.5-2.7) library and command-line tool for streamlining the use of SSH for application deployment or systems administration tasks.
- ▶ <http://www.fabfile.org/>
- ▶ Like a parallel shell, but better.

What is fabric

- ▶ Fabric is a Python (2.5-2.7) library and command-line tool for streamlining the use of SSH for application deployment or systems administration tasks.
- ▶ <http://www.fabfile.org/>
- ▶ Like a parallel shell, but better.

What is fabric

- ▶ Fabric is a Python (2.5-2.7) library and command-line tool for streamlining the use of SSH for application deployment or systems administration tasks.
- ▶ <http://www.fabfile.org/>
- ▶ Like a parallel shell, but better.

What fabric is *not*

- ▶ It is not a replacement for *Salt*, *Puppet* or *CFEngine3*: it has no client / server setup.
- ▶ It is more in the category of *Chef* for Ruby developers.
- ▶ It has no DSL (as Ansible): you have to write Python code.
- ▶ It is not an interactive tool: you get reported what *stdout* is on the remote hosts.

What fabric is *not*

- ▶ It is not a replacement for *Salt*, *Puppet* or *CFEngine3*: it has no client / server setup.
- ▶ It is more in the category of *Chef* for Ruby developers.
- ▶ It has no DSL (as Ansible): you have to write Python code.
- ▶ It is not an interactive tool: you get reported what *stdout* is on the remote hosts.

What fabric is *not*

- ▶ It is not a replacement for *Salt*, *Puppet* or *CFEngine3*: it has no client / server setup.
- ▶ It is more in the category of *Chef* for Ruby developers.
- ▶ It has no DSL (as Ansible): you have to write Python code.
- ▶ It is not an interactive tool: you get reported what *stdout* is on the remote hosts.

What fabric is *not*

- ▶ It is not a replacement for *Salt*, *Puppet* or *CFEngine3*: it has no client / server setup.
- ▶ It is more in the category of *Chef* for Ruby developers.
- ▶ It has no DSL (as Ansible): you have to write Python code.
- ▶ It is not an interactive tool: you get reported what *stdout* is on the remote hosts.

What fabric is *not*

- ▶ It is not a replacement for *Salt*, *Puppet* or *CFEngine3*: it has no client / server setup.
- ▶ It is more in the category of *Chef* for Ruby developers.
- ▶ It has no DSL (as Ansible): you have to write Python code.
- ▶ It is not an interactive tool: you get reported what *stdout* is on the remote hosts.

BioDec's use of Fabric

We use Fabric for simple things to be run on many hosts:

- ▶ *One shot* tasks.
- ▶ *Throw-away* scripts.
- ▶ For more elaborate or repeatable tasks, as setting up a server with Postfix, we use *Ansible*.

Example: what is the value of a certain parameter in a configuration file, on a given set of hosts ?

BioDec's use of Fabric

We use Fabric for simple things to be run on many hosts:

- ▶ *One shot* tasks.
- ▶ *Throw-away* scripts.
- ▶ For more elaborate or repeatable tasks, as setting up a server with Postfix, we use *Ansible*.

Example: what is the value of a certain parameter in a configuration file, on a given set of hosts ?

BioDec's use of Fabric

We use Fabric for simple things to be run on many hosts:

- ▶ *One shot* tasks.
- ▶ *Throw-away* scripts.
- ▶ For more elaborate or repeatable tasks, as setting up a server with Postfix, we use *Ansible*.

Example: what is the value of a certain parameter in a configuration file, on a given set of hosts ?

BioDec's use of Fabric

We use Fabric for simple things to be run on many hosts:

- ▶ *One shot* tasks.
- ▶ *Throw-away* scripts.
- ▶ For more elaborate or repeatable tasks, as setting up a server with Postfix, we use *Ansible*.

Example: what is the value of a certain parameter in a configuration file, on a given set of hosts ?

BioDec's use of Fabric

We use Fabric for simple things to be run on many hosts:

- ▶ *One shot* tasks.
- ▶ *Throw-away* scripts.
- ▶ For more elaborate or repeatable tasks, as setting up a server with Postfix, we use *Ansible*.

Example: what is the value of a certain parameter in a configuration file, on a given set of hosts ?

Example

1. This code fragment is in a file called `misc.py`:

```
@task
def report_postfix_relayhost():
    """Use postconf to get info about the relayhost."""
    run('postconf | grep relayhost')
```

2. The task is executed with the following syntax:

```
fab -f ./ misc.report_postfix_relayhost
```

3. This is what happens (since the host was not defined, it is asked interactively):

```
No hosts found. Please specify (single) host string for connection: devops.ovh.biodec.com
[devops.ovh.biodec.com] run: postconf | grep relayhost
[devops.ovh.biodec.com] out: address_verify_relayhost = $relayhost
[devops.ovh.biodec.com] out: address_verify_sender_dependent_relayhost_maps = $sender_dependent_relayhost_maps
[devops.ovh.biodec.com] out: empty_address_relayhost_maps_lookup_key = <>
[devops.ovh.biodec.com] out: relayhost = git.ovh.biodec.com
[devops.ovh.biodec.com] out: sender_dependent_relayhost_maps =
[devops.ovh.biodec.com] out:
Done.
Disconnecting from devops.ovh.biodec.com... done.
```

Fabric is a lot more

- ▶ It can be used through fab and fabfiles, or as a library.
- ▶ It has a lot of built-ins: run, sudo, put, get, cd, . . .
- ▶ It has many options (skip unreachable hosts, ignore known hosts warning, etcetera).

Fabric is a lot more

- ▶ It can be used through fab and fabfiles, or as a library.
- ▶ It has a lot of built-ins: run, sudo, put, get, cd, . . .
- ▶ It has many options (skip unreachable hosts, ignore known hosts warning, etcetera).

Fabric is a lot more

- ▶ It can be used through fab and fabfiles, or as a library.
- ▶ It has a lot of built-ins: run, sudo, put, get, cd, . . .
- ▶ It has many options (skip unreachable hosts, ignore known hosts warning, etcetera).

Fabric is a lot more

- ▶ It can be used through fab and fabfiles, or as a library.
- ▶ It has a lot of built-ins: run, sudo, put, get, cd, . . .
- ▶ It has many options (skip unreachable hosts, ignore known hosts warning, etcetera).

Upgrade shellshock

```
def upgrade_bash():
    """Just upgrade bash."""

    codename=run('lsb_release -c').split()[-1]
    uname_arch=run('uname -i')
    if ("etch" in codename) and ("x86_64" not in uname_arch):
        put('files/*2.05b-26.3_i386.deb', '/tmp/')
        run('DEBIAN_FRONTEND=noninteractive dpkg -i \
            /tmp/bash_2.05b-26.3_i386.deb \
            /tmp/bash-builtins_2.05b-26.3_i386.deb')
        run('echo "bash hold" | dpkg --set-selections')
        run('echo "bash-builtins hold" | dpkg --set-selections')
    elif ("lenny" in codename) and ("x86_64" not in uname_arch):
        put('files/*3.1.dfsg-8.2_i386.deb', '/tmp/')
        run('DEBIAN_FRONTEND=noninteractive dpkg -i \
            /tmp/bash_3.1.dfsg-8.2_i386.deb \
            /tmp/bash-builtins_3.1.dfsg-8.2_i386.deb')
        run('echo "bash hold" | dpkg --set-selections')
        run('echo "bash-builtins hold" | dpkg --set-selections')
    else:
        run('DEBIAN_FRONTEND=noninteractive apt-get -o \
            Acquire::http::Pipeline-Depth="0" -y update')
        run('DEBIAN_FRONTEND=noninteractive apt-get -o \
            Acquire::http::Pipeline-Depth="0" install bash')
```

Index

Shellshock

Fabric in few words

Know your infrastructure

Happy end !

The missing part

- ▶ You've got the advisory, and you are ready to react: *good !*
- ▶ You wrote the fabfile to upgrade the infrastructure: *very good !*
- ▶ You *even* tested it on some spare hosts: *awesome !*
- ▶ Now . . . how do we get the list of the hosts that need to be updated ?

The missing part

- ▶ You've got the advisory, and you are ready to react: *good !*
- ▶ You wrote the fabfile to upgrade the infrastructure: *very good !*
- ▶ You *even* tested it on some spare hosts: *awesome !*
- ▶ Now . . . how do we get the list of the hosts that need to be updated ?

The missing part

- ▶ You've got the advisory, and you are ready to react: *good !*
- ▶ You wrote the fabfile to upgrade the infrastructure: *very good !*
- ▶ You *even* tested it on some spare hosts: *awesome !*
- ▶ Now ... how do we get the list of the hosts that need to be updated ?

The missing part

- ▶ You've got the advisory, and you are ready to react: *good !*
- ▶ You wrote the fabfile to upgrade the infrastructure: *very good !*
- ▶ You *even* tested it on some spare hosts: *awesome !*
- ▶ Now ... how do we get the list of the hosts that need to be updated ?

The missing part

- ▶ You've got the advisory, and you are ready to react: *good !*
- ▶ You wrote the fabfile to upgrade the infrastructure: *very good !*
- ▶ You *even* tested it on some spare hosts: *awesome !*
- ▶ Now . . . how do we get the list of the hosts that need to be updated ?

The infrastructure

- ▶ At the time we (BioDec) managed around 350 servers (physical, virtual, few network appliances and routers) for a whole of around 8800 services — now they are fewer.
- ▶ We manage the infrastructure with a FTE of 1.5 person/year, so 1 person for 200 servers.
- ▶ We use Check_MK Multisite, with Livestatus, and few customizations to put Facter data into a federation of Couchdb servers (that act as a sort of CMDB).

The infrastructure

- ▶ At the time we (BioDec) managed around 350 servers (physical, virtual, few network appliances and routers) for a whole of around 8800 services — now they are fewer.
- ▶ We manage the infrastructure with a FTE of 1.5 person/year, so 1 person for 200 servers.
- ▶ We use Check_MK Multisite, with Livestatus, and few customizations to put Facter data into a federation of Couchdb servers (that act as a sort of CMDB).

The infrastructure

- ▶ At the time we (BioDec) managed around 350 servers (physical, virtual, few network appliances and routers) for a whole of around 8800 services — now they are fewer.
- ▶ We manage the infrastructure with a FTE of 1.5 person/year, so 1 person for 200 servers.
- ▶ We use Check_MK Multisite, with Livestatus, and few customizations to put Facter data into a federation of Couchdb servers (that act as a sort of CMDB).

The infrastructure

- ▶ At the time we (BioDec) managed around 350 servers (physical, virtual, few network appliances and routers) for a whole of around 8800 services — now they are fewer.
- ▶ We manage the infrastructure with a FTE of 1.5 person/year, so 1 person for 200 servers.
- ▶ We use Check_MK Multisite, with Livestatus, and few customizations to put Facter data into a federation of Couchdb servers (that act as a sort of CMDB).

Query the infrastructure


- ▶

```
$ ./biodecstatus -c biodecstatus.conf -q ssh -s biodec
[ { 'couchdb': { 'queries': { }, 'results': { } } },
  { 'livestatus': { 'queries': { 'query': 'ssh' },
                    'results': { 'query=ssh': [ 'alberto-dev.bo.biodec.com',
                                                'beholder.bo.biodec.com',
                                                <...>
                                                'tova22.bo.biodec.com',
                                                'villa.bo.biodec.com',
                                                'yanez.biodec.com',
                                                'yurta.bo.biodec.com',
                                                'zapata.bo.biodec.com' ] } } } ]
```
- ▶

```
$ time ./biodecstatus -c biodecstatus.conf -q ssh -s all
```

returns 158 hits (in an average on 1.5 minutes)

Happy end !

- 

We contained the shellshock !

There were few rough spots:

- ▶ Linux distributions updated again the bash packages in the following days (at the end there were *six* CVE): we had to re-run the script many times,
- ▶ the hosts lists had to be curated from false positives (*i.e.* QNAPs, Openwrt, even few M\$ boxes with sshd running from Cygwin),
- ▶ how do you test the upgrade when you have *one* server of that kind (Debian Sarge, installed while Amenhotep IV ruled, circa 1360 BC) ?

We contained the shellshock !

There were few rough spots:

- ▶ Linux distributions updated again the bash packages in the following days (at the end there were *six* CVE): we had to re-run the script many times,
- ▶ the hosts lists had to be curated from false positives (*i.e.* QNAPs, Openwrt, even few M\$ boxes with sshd running from Cygwin),
- ▶ how do you test the upgrade when you have *one* server of that kind (Debian Sarge, installed while Amenhotep IV ruled, circa 1360 BC) ?

Insert here your favourite funny picture of a lot of cute cats in triumph.

Questions ?

- ▶ Thanks for paying attention.
- ▶ BioDec is hiring: ping me if you are interested.
- ▶ Follow me on Twitter as @gaunilone, or @biodec, or @incontrodevops.