Computer Systems Security
exercises

Henrik Lund Kramshoej

hlk@zencurity.com

May 27, 2019

Contents

CONTENTS

Preface

This material is prepared for use in *Computer Systems Security workshop* and was prepared by Henrik Lund Kramshoej, http://www.zencurity.com . It describes the networking setup and applications for trainings and workshops where hands-on exercises are needed.

Further a presentation is used which is available as PDF from kramse@Github Look for system-security-exercisesin the repo security-courses.

These exercises are expected to be performed in a training setting with network connected systems. The exercises use a number of tools which can be copied and reused after training. A lot is described about setting up your workstation in the repo

https://github.com/kramse/kramse-labs

Prerequisites

This material expect that participants have a working knowledge of TCP/IP from a user perspective. Basic concepts such as web site addresses and email should be known as well as IP-addresses and common protocols like DHCP.

Have fun and learn

Exercise content

Most exercises follow the same procedure and has the following content:

- **Objective:** What is the exercise about, the objective

- **Purpose:** What is to be the expected outcome and goal of doing this exercise

- **Suggested method:** suggest a way to get started

- **Hints:** one or more hints and tips or even description how to do the actual exercises

- **Solution:** one possible solution is specified

- **Discussion:** Further things to note about the exercises, things to remember and discuss

Please note that the method and contents are similar to real life scenarios and does not detail every step of doing the exercises. Entering commands directly from a book only teaches typing, while the exercises are designed to help you become able to learn and actually research solutions.

Exercise 1

Download Kali Linux Revealed (KLR) Book 10 min

Kali Linux Revealed Mastering the Penetration Testing Distribution

Objective:
We need a Kali Linux for running tools during the course. This is open source, and the developers have released a whole book about running Kali Linux.

This is named Kali Linux Revealed (KLR)

Purpose:
We need to install Kali Linux in a few moments, so better have the instructions ready.

Suggested method:
Create folders for educational materials. Go to https://www.kali.org/download-kali-linux-revealed-book/ Read and follow the instructions for downloading the book.

Solution:
When you have a directory structure for download for this course, and the book KLR in PDF you are done.

Discussion:
Linux is free and everywhere. The tools we will run in this course are made for Unix, so they run great on Linux.

Kali Linux is a free pentesting platform, and probably worth more than $10.000

The book KLR is free, but you can buy/donate, and I recommend it.

Exercise 2

Check your Kali VM, run Kali Linux 30 min

Objective:
Make sure your virtual machine is in working order.

We need a Kali Linux for running tools during the course.

Purpose:
If your VM is not installed and updated we will run into trouble later.

Suggested method:
Go to https://github.com/kramse/kramse-labs/

Read the instructions for the setup of a Kali VM.

Hints:
If you allocate enough memory and disk you wont have problems.

Solution:
When you have a updated virtualisation software and Kali Linux, then we are good.

Discussion:
Linux is free and everywhere. The tools we will run in this course are made for Unix, so they run great on Linux.

Kali Linux includes many hacker tools and should be known by anyone working in infosec.

Exercise 3

Check your Debian VM 10 min

Objective:
Make sure your virtual Debian 9 machine is in working order.

We need a Debian 9 Linux for running a few extra tools during the course.

This is a bonus exercise - only one Debian is needed per team.

Purpose:
If your VM is not installed and updated we will run into trouble later.

Suggested method:
Go to https://github.com/kramse/kramse-labs/

Read the instructions for the setup of a Kali VM.

Hints:

Solution:
When you have a updated virtualisation software and Kali Linux, then we are good.

Discussion:
Linux is free and everywhere. The tools we will run in this course are made for Unix, so they run great on Linux.

Exercise 4

Investigate /etc 10 min

Objective:
We will investigate the /etc directory on Linux

We need a Debian 9 Linux and a Kali Linux, to compare

Purpose:
Start seeing example configuration files, including:

- User database `/etc/passwd` and `/etc/group`

- The password database `/etc/shadow`

Suggested method:
Boot your Linux VMs, log in

Investigate permissions for the user database files `passwd` and `shadow`

Hints:
Linux has many tools for viewing files, the most efficient would be less.

```
hlk@debian:~$ cd /etc
hlk@debian:/etc$ ls -l shadow passwd
-rw-r--r-- 1 root root   2203 Mar 26 17:27 passwd
-rw-r----- 1 root shadow 1250 Mar 26 17:27 shadow
hlk@debian:/etc$ ls
... all files and directories shown, investigate more if you like
```

Showing a single file: `less /etc/passwd` and press q to quit

Showing multiple files: `less /etc/*` then :n for next and q for quit

```
Trying reading the shadow file as your regular user:
user@debian-9-lab:/etc$ cat /etc/shadow
cat: /etc/shadow: Permission denied
```

Why is that? Try switching to root, using su or sudo, and redo the command.

Solution:
When you have seen the most basic files you are done.

Discussion:
Linux is free and everywhere. The tools we will run in this course are made for Unix, so they run great on Linux.

Exercise 5

Discover active systems ping sweep 10 min

Objective:
Use nmap to discover active systems

Purpose:
Know how to use nmap to scan networks for active systems.

Suggested method:
Try different scans,

- Ping sweep to find active systems

- Port sweeps to find active systems with specific ports

Hints:
Try nmap in sweep mode - and you may run this from Zenmap

Solution:
Use the command below as examples:

- Ping sweep `nmap -sP 10.0.45.*`

- Port sweeps `nmap -p 80 10.0.45.*`

Discussion:
Quick scans quickly reveal interesting hosts, ports and services

Also now make sure you understand difference between single host scan 10.0.45.123/32, a whole subnet /24 250 hosts 10.0.45.0/24 and other more advanced targeting like 10.0.45.0/25 and 10.0.45.1-10

Exercise 6

Execute nmap TCP and UDP port scan 20 min

Objective:
Use nmap to discover important open ports on active systems

Purpose:
Finding open ports will allow you to find vulnerabilities on these ports.

Suggested method:
Use `nmap -p 1-1024 server` to scan the first 1024 TCP ports and use Nmap without ports. What is scanned then?

Try to use `nmap -sU` to scan using UDP ports, not really possible if a firewall is in place.

If a firewall blocks ICMP you might need to add `-Pn` to make nmap scan even if there are no Ping responses

Hints:
Sample command: `nmap -Pn -sU -p1-1024 server` UDP port scanning 1024 ports without doing a Ping first

Solution:
Discover some active systems and most interesting ports, which are 1-1024 and the built-in list of popular ports.

Discussion:
There is a lot of documentation about the nmap portscanner, even a book by the author of nmap. Make sure to visit http://www.nmap.org

TCP and UDP is very different when scanning. TCP is connection/flow oriented and requires a handshake which is very easy to identify. UDP does not have a handshake and most applications will not respond to probes from nmap. If there is no firewall the operating system will respond to UDP probes on closed ports - and the ones that do not respond must be open.

When doing UDP scan on the internet you will almost never get a response, so you cannot tell open (not responding services) from blocked ports (firewall drop packets). Instead try using specific service programs for the services, sample program could be `nsping` which sends DNS packets, and will often get a response from a DNS server running on UDP port 53.

Exercise 7

Perform nmap OS detection 10 min

Objective:
Use nmap OS detection and see if you can guess the brand of devices on the network

Purpose:
Getting the operating system of a system will allow you to focus your next attacks.

Suggested method:
Look at the list of active systems, or do a ping sweep.

Then add the OS detection using the option –O

Better to use -A all the time, includes even more scripts and advanced stuff See the next exercise.

Hints:
The nmap can send a lot of packets that will get different responses, depending on the operating system. TCP/IP is implemented using various constants chosen by the implementors, they have chosen different standard packet TTL etc.

Solution:
Use a command like `nmap –O –p1–100 10.0.45.45` or `nmap –A –p1–100 10.0.45.45`

Discussion:
nmap OS detection is not a full proof way of knowing the actual operating system, but in most cases in can detect the family and in some cases it can identify the exact patch level of the system.

Exercise 8

Run Armitage - Hail Mary 30min

Objective:
Try hacking using a graphical program, see how quick and easy it can be.

Purpose:
Show that when a vulnerability exist attacks can be quick and easy.

Suggested method:

Running Armitage as a gui on top of Metasploit is the easiest way to do this.

1. Boot up Kali Linux

2. Boot up Metasploitable - from ISO
   There may be a couple of systems already running this.

3. Run Armitage Hail-Mary against Metasploitable

4. Note which succeeded, describe those attacks that succeeded in relation to MITRE ATT&CK framework

Hints:
Running Metasploit against Metasploitable - which is a vulnerable system - should result in multiple vulnerabilities exploited.

Each of these may have different characteristics.

We are aiming at:

- Vulnerable application - root access

- Vulnerable application - non-root access, would need privilege escalation

- Bad password allowing *Brute Force* access, msfadmin/msfadmin - see also *Valid Accounts*

Solution:
When you have exploited and mapped at least one vulnerability you are done, but should spend more time.

Discussion:
Do we need these frameworks? What are the benefits? - can we become product blind - so we only see what these framework cover.

11

Exercise 9

SELinux Introduction up to 60min

Objective:
Check out the SELinux system
https://www.debian.org/doc/manuals/debian-handbook/sect.selinux.en.html

and the setup instructions at:
https://wiki.debian.org/SELinux/Setup

(Not working right now - Create a secret file, that you can read, but root cant.)

Purpose:
Everybody reads about Discretionary Access Control (DAC) and Mandatory Access Control (MAC) but few realize that Linux implements it.

Suggested method:
Try enabling and disabling the policies in your Debian VM.

First install prerequisites - approx 75MB download on my system:
```
apt-get install selinux-basics selinux-policy-default auditd
```

Then run activation of SELinux:
```
selinux-activate
```

```
root@debian-9-lab:~# selinux-activate
Activating SE Linux
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-4.9.0-9-amd64
Found initrd image: /boot/initrd.img-4.9.0-9-amd64
Found linux image: /boot/vmlinuz-4.9.0-8-amd64
Found initrd image: /boot/initrd.img-4.9.0-8-amd64
done
SE Linux is activated.  You may need to reboot now.
root@debian-9-lab:~#
```

Perform the reboot, `shutdown -r now` then check again.

Not enabled will show this, try again:

```
root@debian-9-lab:~# sestatus
SELinux status:                 disabled
```

Enabled, but not the current mode and mode from config file discrepancy:

```
root@debian:~# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             default
Current mode:                    enforcing
Mode from config file:           permissive
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Max kernel policy version:      30
```

While playing I had changed the mode temporarily to *enforcing*! Next reboot would make SELinux run in the more *permissive* mode

9.0.1   Part 2 - do this when SELinux is enabled

Create a directory and a test file:

```
root@debian:~# setenforce 0   // set mode permissive!
root@debian:~# cd
root@debian:~# mkdir /etc/private
root@debian:~# echo "hey" > /etc/private/README
root@debian:~# cat /etc/private/README
hey
root@debian:~#
```

Root can read the file, yay!

Copy example files:

```
cp -r /usr/share/doc/selinux-policy-dev/examples .
cd examples/
```

Create a file myprivate.te with this content:

```
policy_module(myprivate, 1.0)

########################################
#
# Declarations
#
type etc_private_t;
```

```
fs_associate(etc_private_t)

type sysadm_t;
type sysadm_exec_t;

userdom_admin_user_template(sysadm_t)

allow sysadm_t etc_private_t:dir file relabelto;
```

Note last line is missing a sysadm domain, does not work.

Then compile using this: `make myprivate.pp`

```
root@debian:~/examples# make myprivate.pp
Compiling default myprivate module
/usr/bin/checkmodule:  loading policy configuration from tmp/myprivate.tmp
/usr/bin/checkmodule:  policy configuration loaded
/usr/bin/checkmodule:  writing binary representation (version 17) to tmp/myprivate.mod
Creating default myprivate.pp policy package
rm tmp/myprivate.mod.fc tmp/myprivate.mod
root@debian:~/examples#
```

then it should have been possible to enable/disable enforcing mode, and see the file becoming unreadable - even by root.

Something is wrong, when enabling enforcing mode, the chcon command fails:

```
root@debian:~/examples# setenforce 1
root@debian:~/examples# chcon -R -t etc_private_t /etc/private/README
chcon: failed to change context of '/etc/private/README' to 'system_u:object_r:etc_private_t:s
root@debian:~/examples# chcon -R -t etc_private_t /etc/private
chcon: failed to change context of 'README' to 'system_u:object_r:etc_private_t:s0': Invalid a
chcon: failed to change context of '/etc/private' to 'system_u:object_r:etc_private_t:s0': Inv

root@debian:~/examples# setenforce 0
root@debian:~/examples# chcon -R -t etc_private_t /etc/private/README
root@debian:~/examples#
// When Linux returns to the command prompt without messages no errors were observed
```

So SELinux IS preventing us from doing it :-D

this example is in parts based on this blog post:
http://blog.siphos.be/2015/07/restricting-even-root-access-to-a-folder/

Hints:
Keeping SELinux enabled may NOT be a good idea, since some tools may not work correctly, until policies are downloaded, written or installed.

Temporarily disable SELinux:
```
echo 0 > /sys/fs/selinux/enforce
```

Temporarily enable SELinux:
```
echo 1 > /sys/fs/selinux/enforce
```

or use the command `setenforce 0` or `setenforce 1`

The main config for setting permissive or enforcing mode is `/etc/selinux/config`:

```
root@debian-9-lab:~# cat /etc/selinux/config
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of these two values:
# default - equivalent to the old strict and targeted policies
# mls     - Multi-Level Security (for military and educational use)
# src     - Custom policy built from source
SELINUXTYPE=default

# SETLOCALDEFS= Check local definition changes
SETLOCALDEFS=0
```

Solution:
When you have enabled and seen the commands used, you are done.

It is easy to have multiple hours disappear when working with SELinux.

Discussion:
Yes, the root user can disable the SELinux protection :-D

I had Firefox crash at least once during this exercise, so beware - fancy and bigger applications may crash when using this!

Exercise 10

Example AUPs up to 30min

Objective:
See real world high level policies

Purpose:
When writing your first policy it may be hard to know what to include. Starting from an example is often easier.

Suggested method:
Find your AUP for the ISPs we use, you use, your company uses.

Hints:
Policies for different environments are often very different in scope and goals.

Book mentions military and commercial, but an ISP, University and a commercial enterprise have very different methods and requirements.

Example, how do you handle BYOD Bring your own devices, University you expect students to bring them, in a secure enterprise only company devices may be allowed.

Solution:
When you have seen at least two different policies you are done.

Discussion:
How do you both write AND create awareness about a policy?

Exercise 11

SYN flooding 101

Objective:
Start a webserver attack using SYN flooding tool hping3.

Purpose:
See how easy it is to produce packets on a network using hacker programs.

The tool we will use is very flexible and can produce ICMP, UDP and TCP using very few options.

```
-1 --icmp
        ICMP  mode,  by  default  hping3  will  send  ICMP echo-request, you can set other ICMP
        type/code using --icmptype --icmpcode options.

-2 --udp
        UDP mode, by default hping3 will send udp to target host's port 0.  UDP header  tunable
        options are the following: --baseport, --destport, --keep.
```

TCP mode is default, so no option needed.

Suggested method:
Connect to the LAB network using Ethernet! Borrow a USB network card if you dont have one.

Start your Kali VM in bridged mode, try a basic TCP flooding attack against the server provided by the instructor, or your own Debian server.

```
hping3 --flood -p 80 10.0.45.12
```

You should see something like this:

```
HPING 10.0.45.12: NO FLAGS are set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 10.0.45.12 hping statistic ---
352339 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Try doing the most common attacks, RTFM hping3:

- ICMP flooding

- UDP flooding, try port 53 and port 123

- TCP flooding, try port 22 or port 80 on your debian perhaps

Hints:
The tool we use can do a lot of different things, and you can control the speed. You can measure at the server being attacked or what you are sending, commonly using ifpps or such programs can help.

This allows you to use the tool to test devices and find the breaking point, which is more interesting than if you can overload, because you always can.

```
-i --interval
        Wait the specified number of  seconds or micro seconds between sending each packet.
        --interval X set wait to X seconds, --interval uX set wait to X micro seconds.  The de-
        fault  is  to  wait one second between each packet. Using hping3 to transfer files tune
        this option is really important in order to increase transfer rate. Even  using  hping3
        to  perform  idle/spoofing  scanning  you should tune this option, see HPING3-HOWTO for
        more information.

--fast Alias for -i u10000. Hping will send 10 packets for second.

--faster
        Alias for -i u1. Faster then --fast ;) (but not as fast as your computer can send pack-
        ets due to the signal-driven design).

--flood
        Sent  packets  as fast as possible, without taking care to show incoming replies.  This
        is ways faster than to specify the -i u0 option.
```

Solution:
When your team has sent +1 million packets per second into the network, from one or two laptops - you are done.

Discussion:
Gigabit Ethernet can send up to 1.4 million packets per second, pps.

There is a presentation about DDoS protection with low level technical measures to implement at
https://github.com/kramse/security-courses/tree/master/presentations/network/introduction-ddos-testing

Exercise 12

RBAC Access permissions on GitHub 30-45min

Objective:
See actual real life example of permissions.

Note: This exercise requires a GitHub account, so make sure your group has one. Maybe do groups of 3-4 for more discussion.

Purpose:
GitHub is a very popular code sharing site.

Suggested method:
Go to GitHub web page:
https://help.github.com/en/articles/access-permissions-on-github

Follow links to other pages, like:
https://help.github.com/en/articles/permission-levels-for-an-organization

Hints:
Some might already have an account on GitHub - maybe work through adding a repository and adding collaborators.

If you have an organisation, even better.

Solution:
When you have discussed GitHub permissions and played with a repository you are done.

Discussion:
The internet is decentralized, but recent years see more centralization - GitHub, DNS Google DNS, Cloudflare.

What are some problems in this?

Exercise 13

SSL/TLS scanners 15 min

Objective:
Try the Online Qualys SSLLabs scanner https://www.ssllabs.com/ Try the command line tool sslscan checking servers - can check both HTTPS and non-HTTPS protocols!

Purpose:
Learn how to efficiently check TLS settings on remote services.

Suggested method:
Run the tool against a couple of sites of your choice.

```
root@kali:~# sslscan --ssl2 web.kramse.dk
Version: 1.10.5-static
OpenSSL 1.0.2e-dev xx XXX xxxx

Testing SSL server web.kramse.dk on port 443
...
  SSL Certificate:
Signature Algorithm: sha256WithRSAEncryption
RSA Key Strength:    2048

Subject:  *.kramse.dk
Altnames: DNS:*.kramse.dk, DNS:kramse.dk
Issuer:   AlphaSSL CA - SHA256 - G2
```

Also run it without `--ssl2` and against SMTPTLS if possible.

Hints:
Originally sslscan is from http://www.titania.co.uk but use the version on Kali, install with apt if not installed.

Solution:
When you can run and understand what the tool does, you are done.

Discussion:
SSLscan can check your own sites, while Qualys SSLLabs only can test from hostname

Exercise 14

Nmap Ikescan IPsec

Objective:
Try Nmap and Ikescan

Purpose:

Suggested method:

Hints:

Solution:

Discussion:

Exercise 15

SSH scanners

Objective:

Try ssh scanners, similar to sslscan and Nmap sshscan

Purpose:

Suggested method:

Hints:

Solution:

Discussion:

22

Exercise 16

Password Cracking

Objective:

Crack your own passwords **Purpose:**

Suggested method:

Hints:

Solution:

Discussion:

Exercise 17

Perform privilege escalation using files 30min

Objective:
Perform a simple privilege escalation attack

Purpose:
Try and test a back door script.

Suggested method:

1. Make a non-privileged user

2. make a system directory writable

3. create root cronjob without path

4. Insert a malicious script as one of the commands from the root cron job

Hints:
A cron job runs scheduled commands. They usually perform cleanup functions, removing old files, doing a backup or similar

In this exercise first try out the malicious commands for creating a back door shell program. Login in as root, then:

```
root@debian:~# rm /tmp/.xxsh
root@debian:~# cp /bin/dash /tmp/.xxsh
root@debian:~# chmod +sw /tmp/.xxsh
```

Then test using a normal user, another window:

```
hlk@debian:~$ /tmp/.xxsh
# id
uid=1000(hlk) gid=1000(hlk) euid=0(root) egid=0(root) groups=0(root),24(cdrom),25(floppy),29
s0:c0.c1023
#
```

The effective user id should be 0 which is root.

When this manual process work. Then automate it, make it into a script like in the book. Imagine if the root user was running automated scripts, and you could add yours to a directory used in the PATH for these automated ones.

This happens in a lot of devices and hosts today.

The main takeaway is that root scripts should ALWAYS have a PATH defined, and ALL directories used by root script should only be writable by root!

Solution:
When you have created the script for doing the shell copy you are done.

Further advanced steps would be to add this into some PATH writable by you, and letting a cron job escalate.

Then do a cron job that uses this command - a cron job running every 5 minutes using the `ls` command and introduce your malicious script by putting it before the real command in the PATH.

Discussion:
This was chosen as I found a similar vulnerability in a professional product, in 2019

Exercise 18

Anti-virus and "endpoint security" 20min

Objective:
Discuss when to use Anti-virus and "endpoint security"

Purpose:
Anti-virus programs have been shown to catch some viruses, useful.

Anti-virus programs have been shown to be insecure programs that also slows down systems, counter-productive and increases target surface and exposure.

Suggested method:
Sit in groups 3-5 – discuss among yourselves. Write down plus and minus for using anti-virus – especially which use-cases should use AV, and which shouldn't.

Hints:
In some cases people have installed AV products for check-mark security, the check-list said to have AV, so we installed a mail scanner on this web server – bad security.

Solution:
When we have done a collected talk and discussion we are done.

Discussion:
I dont use anti-virus products at all. I do use a lot of backup though.

Which is more trust-worthy - a restored system or a system cleaned by random anti-virus program?

Exercise 19

Email Security 2019 up to 45min

Objective:
Talk and plan roll-out of security mechanisms based on DNS records. Domain Name System. Check you personal domain and domain used for work. If you are new to this I suggest experimenting with your own personal domain, or create one.

Purpose:
Make sure everyone attending know about methods to restrict sending of false emails, how to secure this using DNSSEC, SPF, DMARC - DNS based updates to your email domain security

Email security 2019 - Goals

- SPF Sender Policy Framework
  https://en.wikipedia.org/wiki/Sender_Policy_Framework

- DKIM DomainKeys Identified Mail
  https://en.wikipedia.org/wiki/DomainKeys_Identified_Mail

- DMARC Domain-based Message Authentication, Reporting and Conformance
  https://en.wikipedia.org/wiki/DMARC

- DANE DNS-based Authentication of Named Entities
  https://en.wikipedia.org/wiki/DNS-based_Authentication_of_Named_Entities

- Brug allesammen, check efter ændringer!

DNS-based Authentication of Named Entities (dane)

> "TLSA records store hashes of remote server TLS/SSL certificates. The authenticity of a TLS/SSL certificate for a domain name is verified by DANE protocol (RFC 6698). DNSSEC and TLSA validation results are displayer by using several icons."

Suggested method:
Use services on the internet, such as https://internet.nl/ and https://dmarcian.com/ to see current status for your domains.

Hints:
I suggest the following strategy when you implement these methods, if you dare do it right now. If you make a plan.

Basic mail security 2019

1. Implement DNSSEC - turn it on, most likely easy

2. Configure Sender Policy Framework, perhaps only `~all` tilde means soft fail

3. Configure DomainKeys Identified Mail

4. Configure receiving email address for DMARC

5. Configure Domain-based Message Authentication - reject none

Spend some time trying different tools for DMARC reporting. A month or a week, depending on the domain and your users. Github alone has 100s of projects concerned with parsing, reporting and working with DMARC.

Then after some time has passed, and you have reviewed reporting from DMARC, turn it on for real:

1. Configure SPF to disallow with hard fail use `-all` minus

2. Configure DMARC with reject - reject emails not following policy

Advanced mail security 2019

1. Create real certificates for DANE

2. Publish them ☺

Helpful hints from https://blog.apnic.net/2017/01/06/lets-encrypt-dane/

Solution:
When the internet is ridden of falsified spam you are done ☺ - its up to you.

Discussion:
You need to research before making changes to important domains.

If you have domains that *never send email* then add the following SPF and DMARC to avoid misuse.

from my own DSN template for *parked domains*:

```
gdns.template v=spf1 -all 43200
_dmarc.gdns.template v=DMARC1; p=reject; 43200
```