WebGoat.SDWAN.Net
in Depth

Denis Kolegov / @dnkolegov
Oleg Broslavsky / @yalegko

Power of Community - November 8th 2018


SD WAN
NEW HOPE

SD-WAN New Hope

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

- SD-WAN Repository
- SD-WAN Internet Census
- SD-WAN Harvester
- SD-WAN Infiltrator
- SD-WAN Threats (WIP)

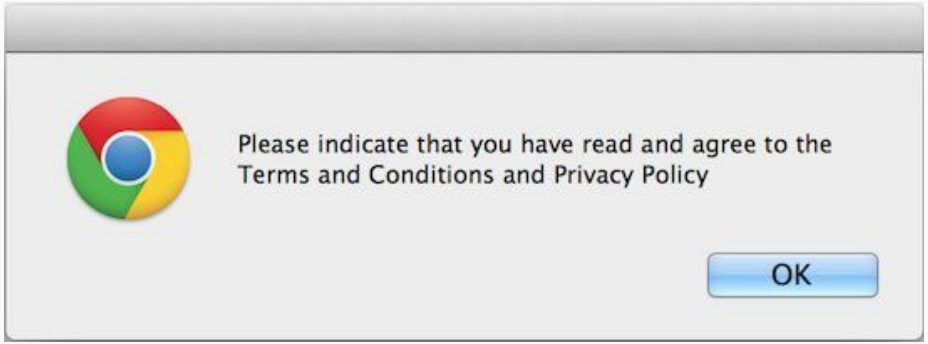
https://github.com/sdnewhop/

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. So, mistakes and bad jokes are all OUR responsibilities
- Actually no one has seen the slides before

Disclaimer (2/2)

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



Intro @Oleg

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

Intro @Denis

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

Why WebGOAT.SDWAN.Net?

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

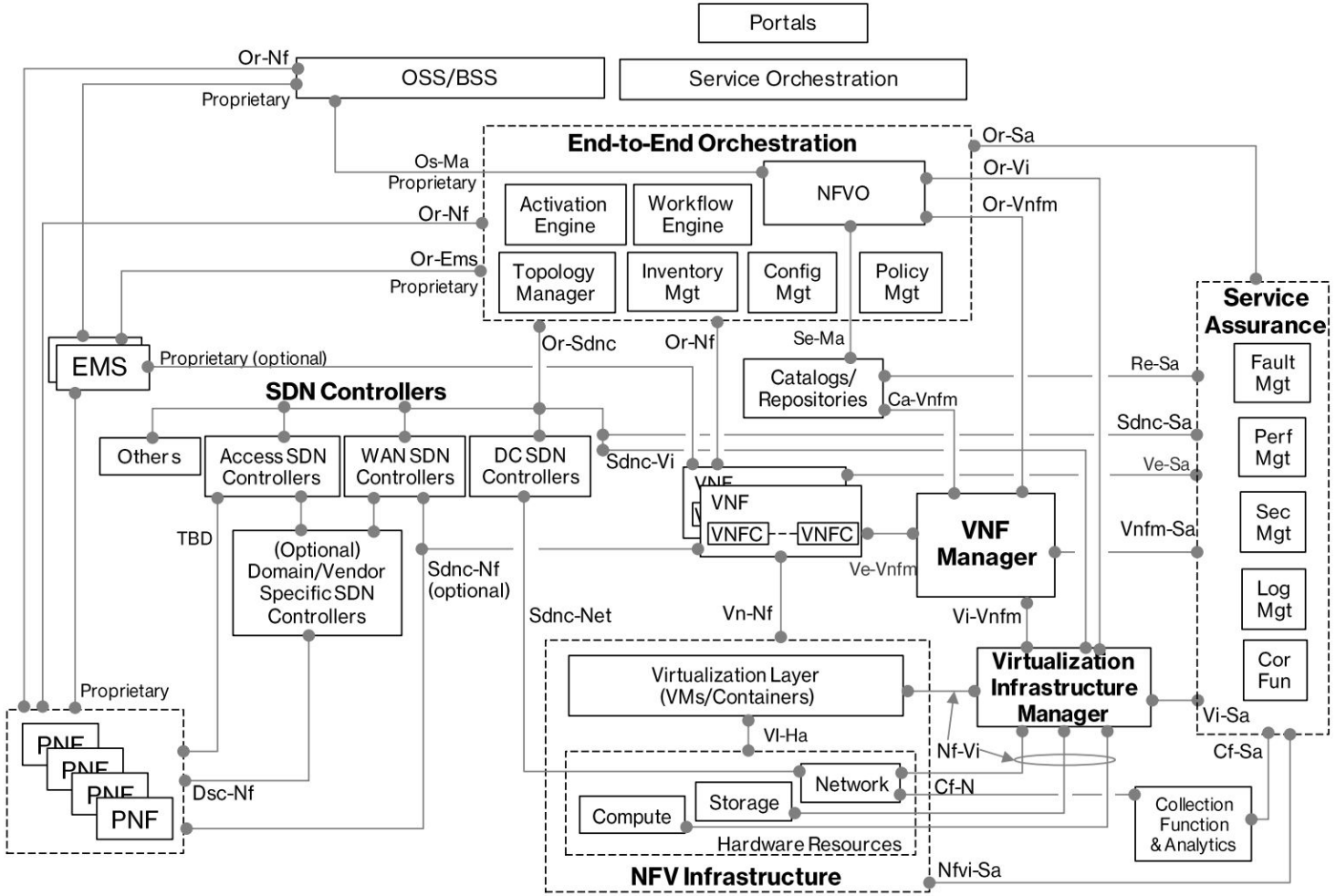
Questions

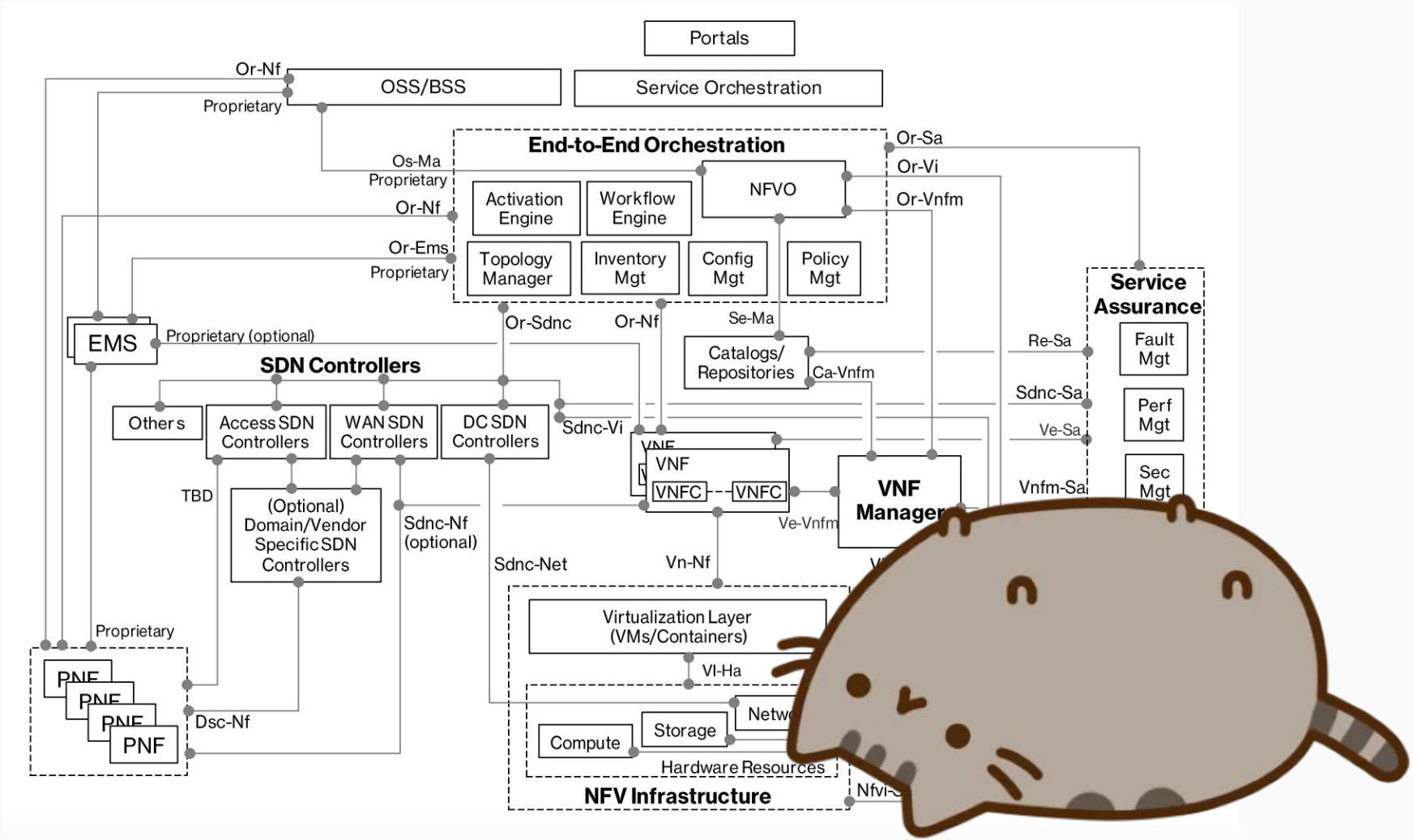
- How many SD-WAN nodes are on the Internet?
- Common security level of SD-WAN products
- Is SD-WAN low-hanging fruit and how low it is?
- How to hack SD-WAN via traditional vulnerabilities?
- Security of SD-WAN specific mechanisms

Agenda

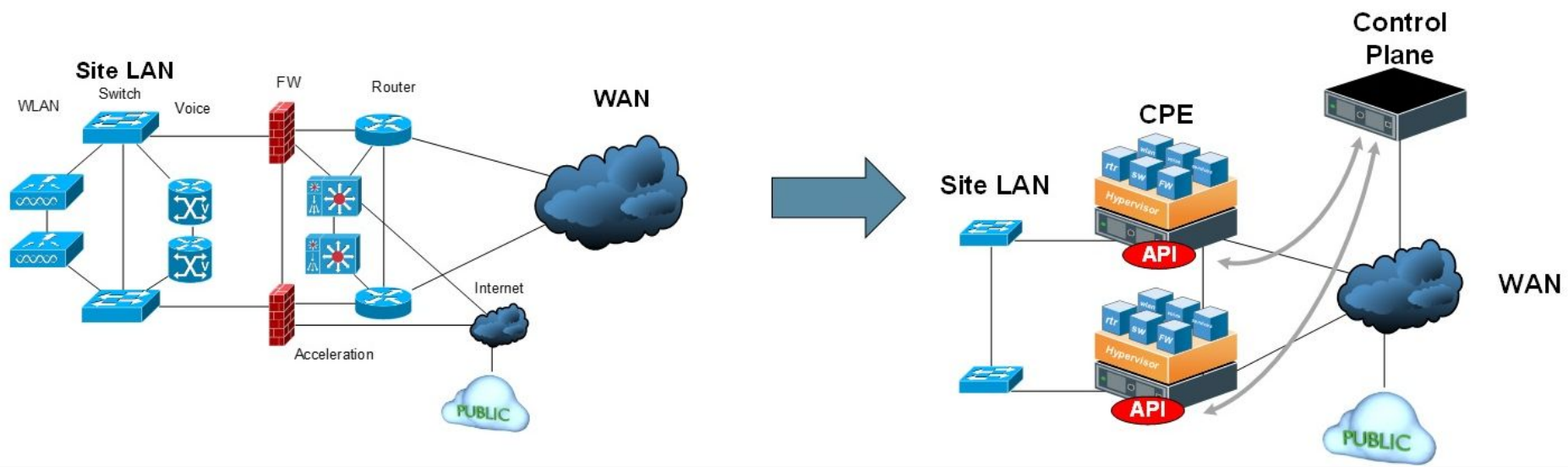
- SD-WAN Essence
- SD-WAN Internet Census
- SD-WAN Vulnerabilities in Practice

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

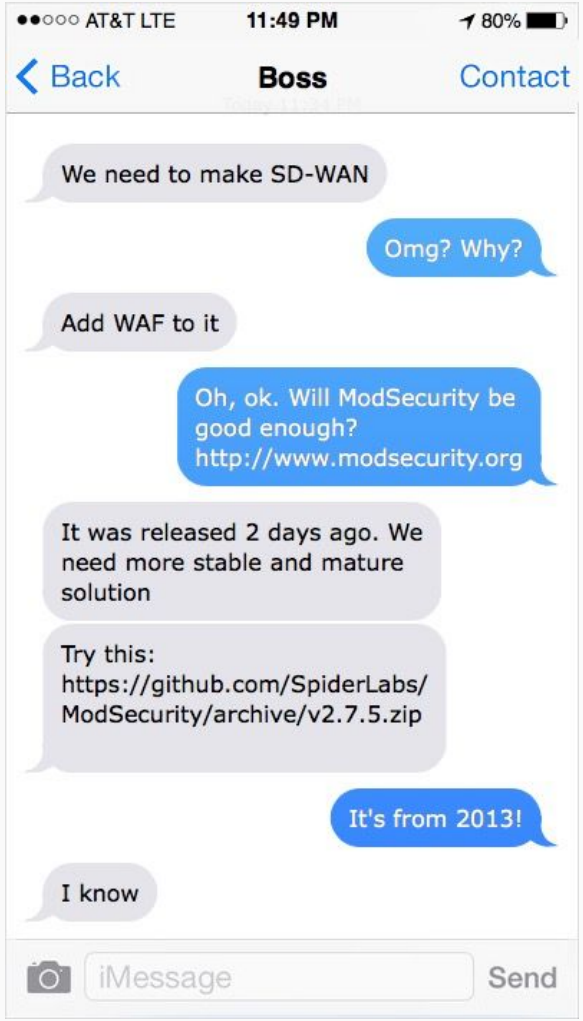
Login

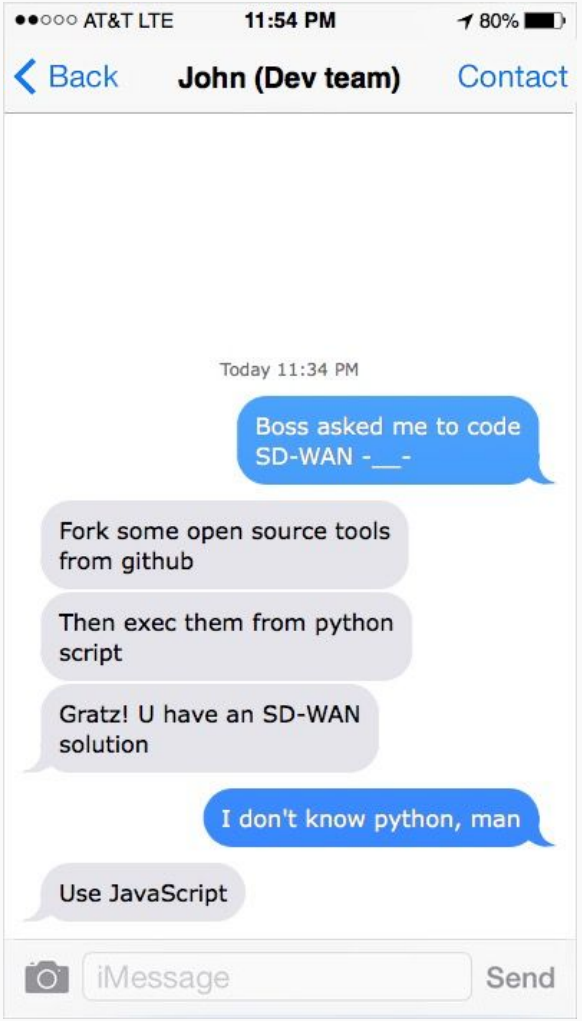
50% 50%

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

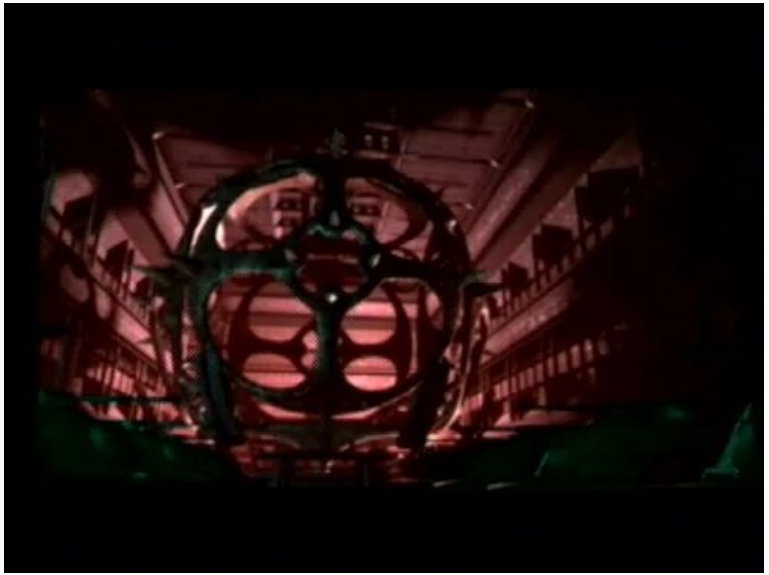
SD-WAN Security

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

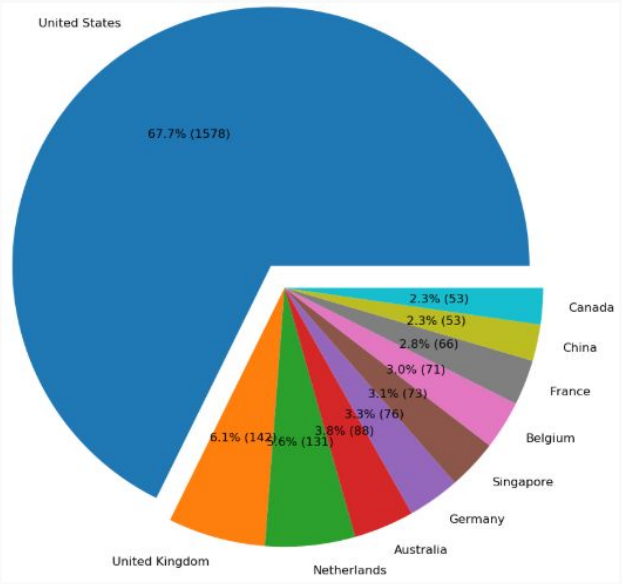
SD-WAN Internet Census

SD-WAN Internet Census

- Best effort approach
- Crafted Shodan and Censys queries
- Found version disclosure patterns
- Developed tools
  - SD-WAN Harvester
  - SD-WAN Infiltrator



SD-WAN Map



Last scan: October, 2018

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

Version Leakage Patterns

```
yalegko:~ $ ssh admin@REDACTED
viptela 17.2.4
admin@REDACTED password:
```

```
<h2 style="margin-top:5px;">FatPipe WARP</h2>
    <h5>9.1.2r142</h5>
```

```
<link href="/br_ui/rdx/core/css/rdx.css?v=9.3.1.35" rel="stylesheet" type="text/css"/>
<link href="/br_ui/app/css/br.css?v=9.3.1.35" rel="stylesheet" type="text/css"/>
```

ZERO NIGHTS 2018

20 - 21 NOVEMBER 2018

SAINT-PETERSBURG

A2 GREEN CONCERT

BUY A TICKET PROGRAM

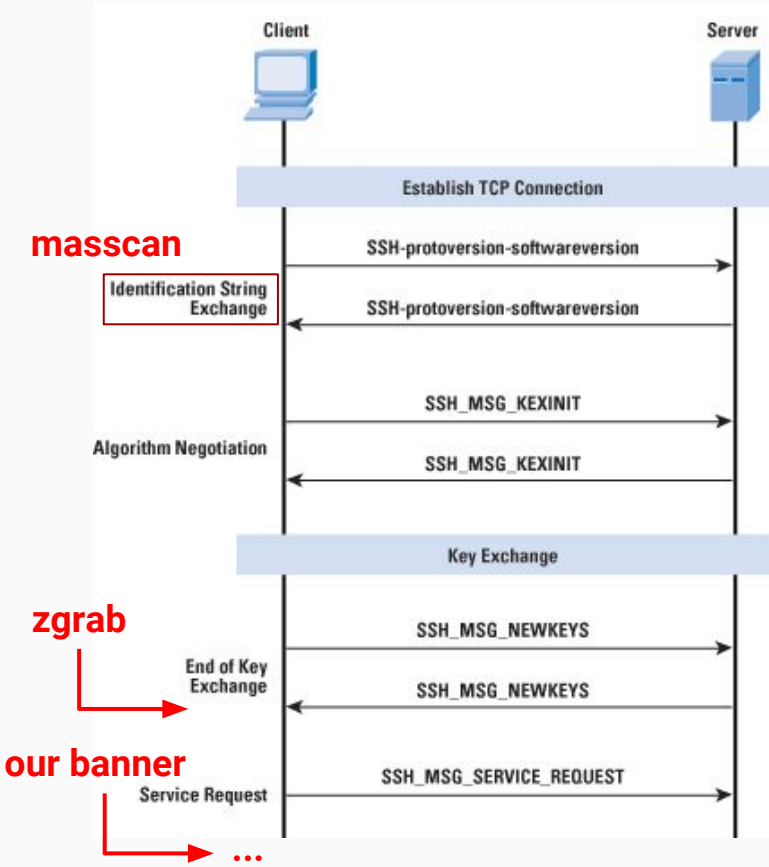
SD-WAN INTERNET CENSUS

NOVEMBER 21, 12:15 | 30MIN | HALL MIR

SSH Fingerprinting

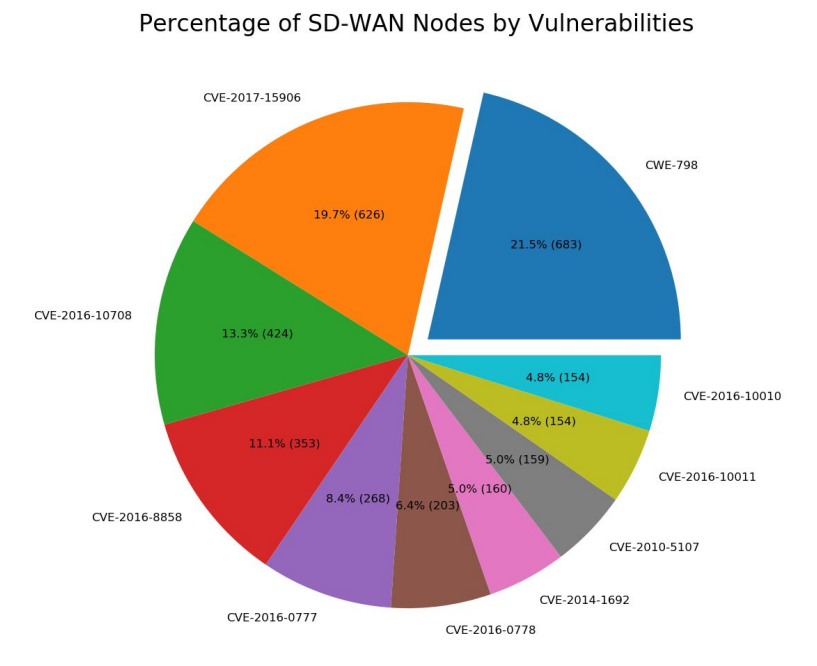
- SD-WAN version in `/etc/issue` message
- It is too complicated even for masscan
  - Implement the rest of SSH protocol
  - Look for another tool
- zgrab does almost everything we need
- Add last steps to the zgrab ssh module
- Use zmap + zgrab for hosts enumeration (feel free to use masscan + zgrab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

https://github.com/sdnewhop/zgrab2



Occasional Findings

SD-WAN OpenSSH Vulnerabilities



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

https://github.com/sdnewhop/sdwan-harvester

TELoIP Orchestrator API Version Disclosure

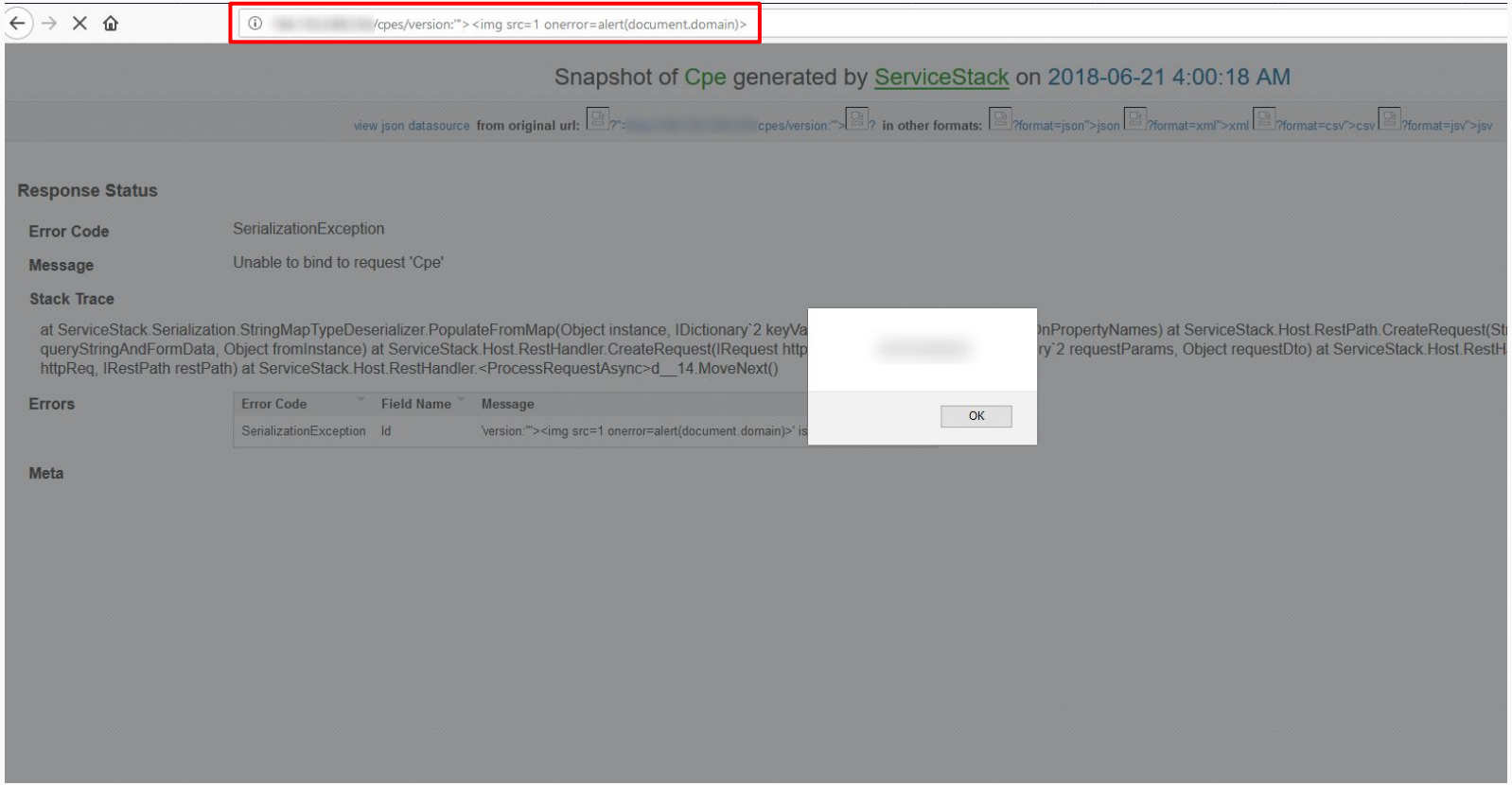
Request: `http://example.com/?debug=requestinfo`

Response:

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

TELoIP Orchestrator API Stack Trace Exposure

TELoIP Orchestrator API XSS



Responsible Disclosure Results

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

TELoIP Support no-reply@teloip.com через gizfxrlz4qwe.41-5rffeaq.na35.bnc.salesforce.com
кому: я

английский > русский Перевести сообщение

Dear Denis Kolegov,

Thank you for submitting your request to TELoIP.

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

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

Thank you,
TELoIP Support
877-783-5647 x2

ref:_00D415rfF._50041aFl2v:ref

No response, but all reported issues were fixed

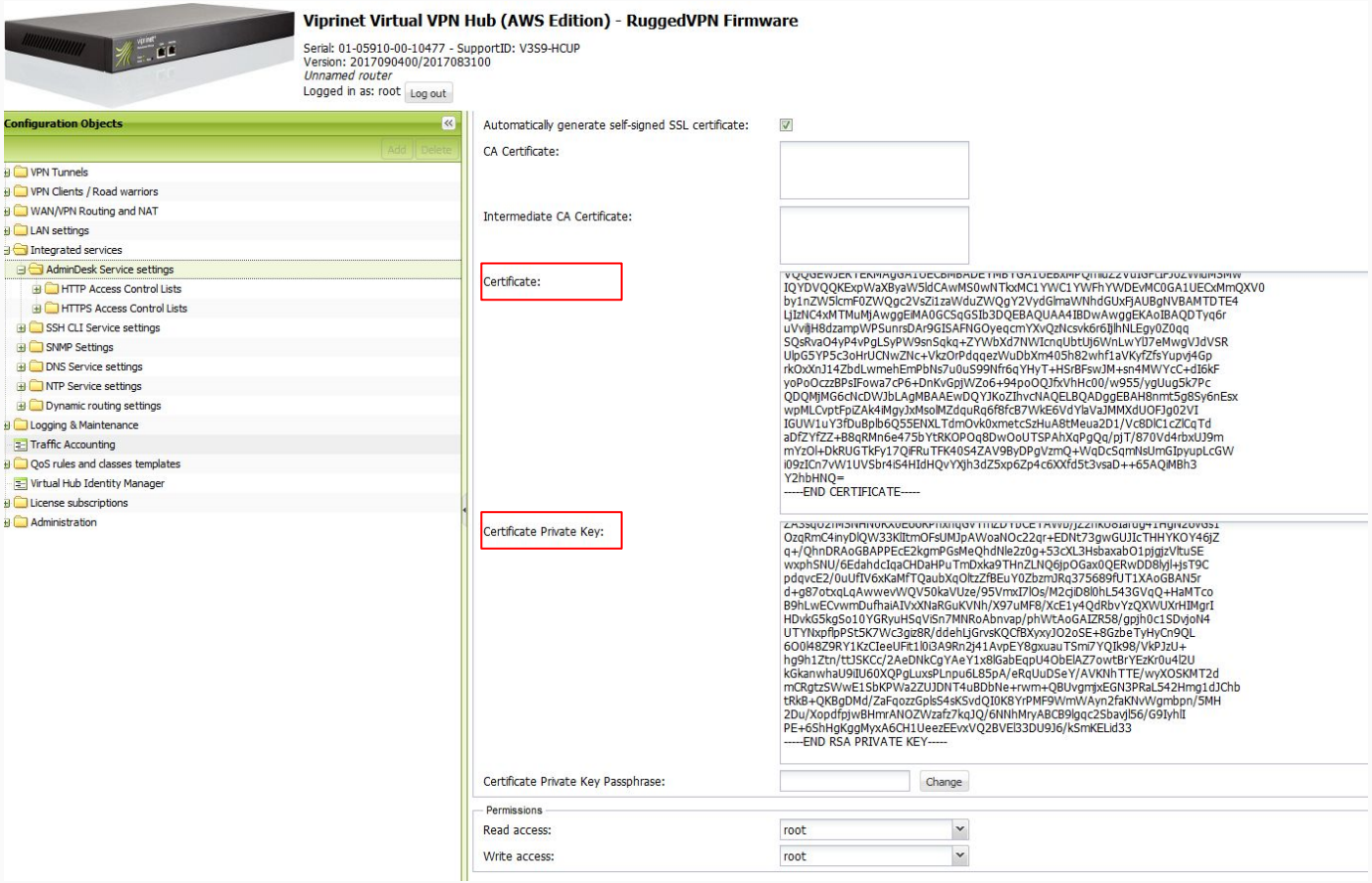
SD-WAN Vulnerabilities in Practice

Viprinet Stored XSS

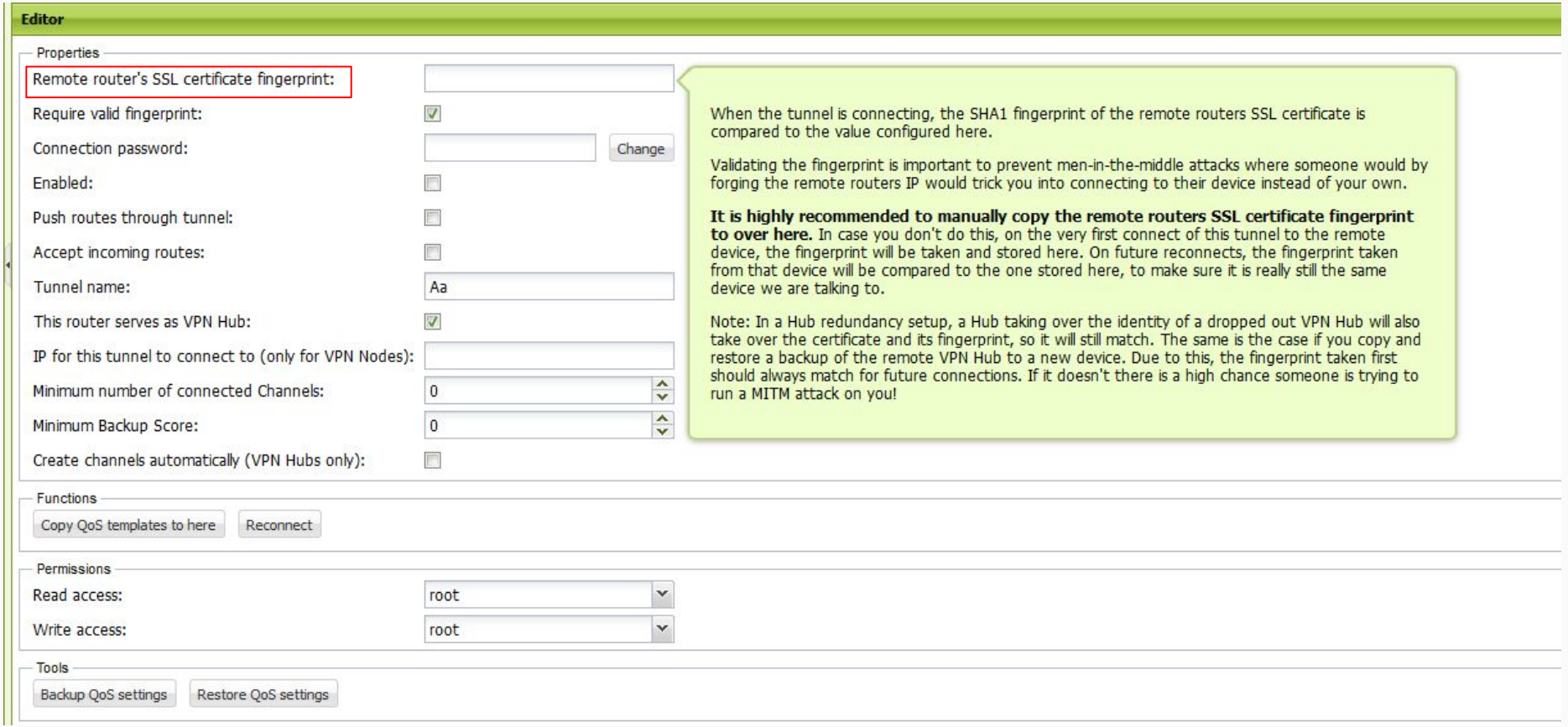
Viprinet XSS

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

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



Viprinet CVE-2014-2045

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

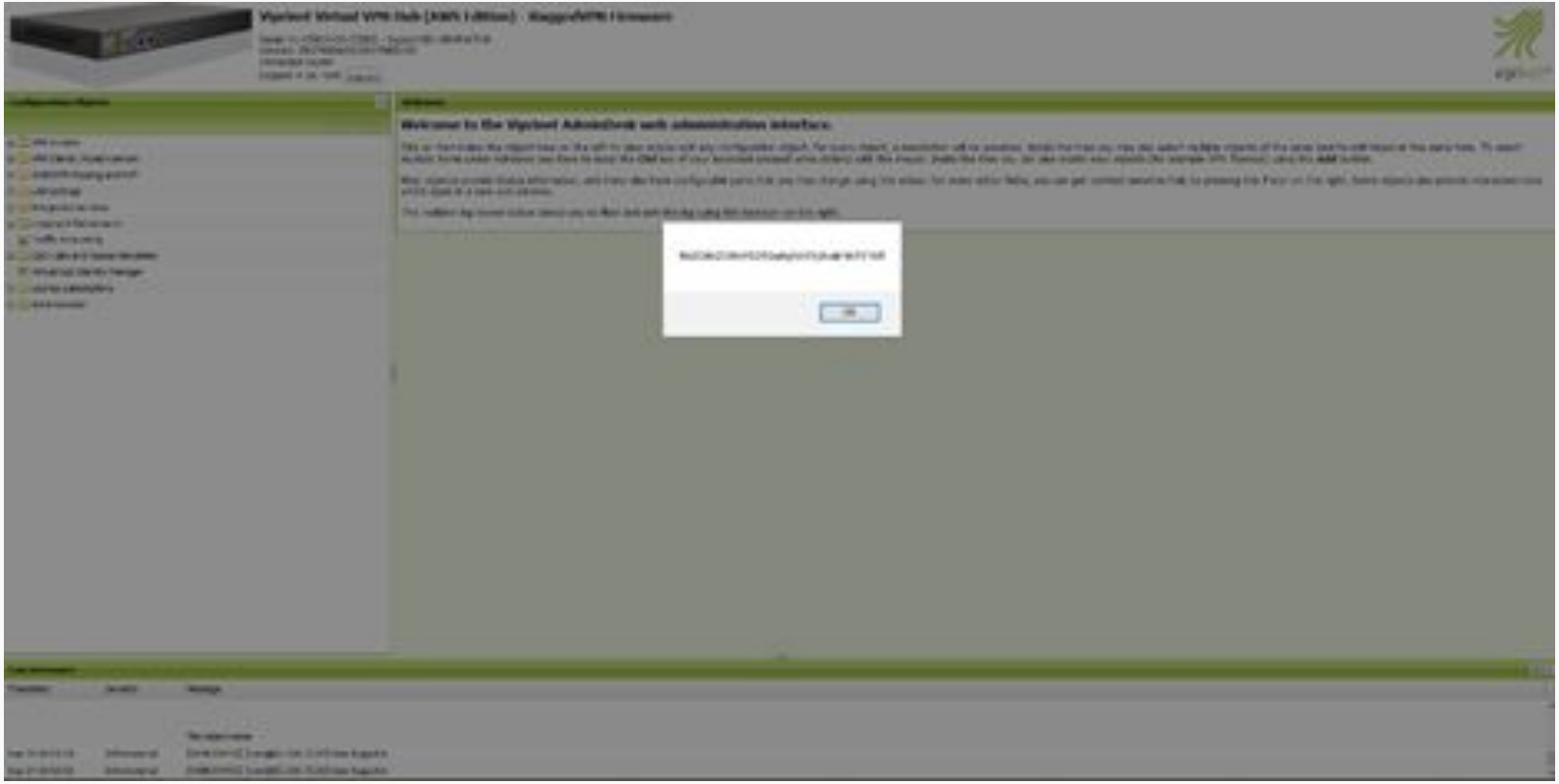
Viprinet Interfaces

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

CLI Commands

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

Viprinet Stored XSS via CLI



Responsible Disclosure Results

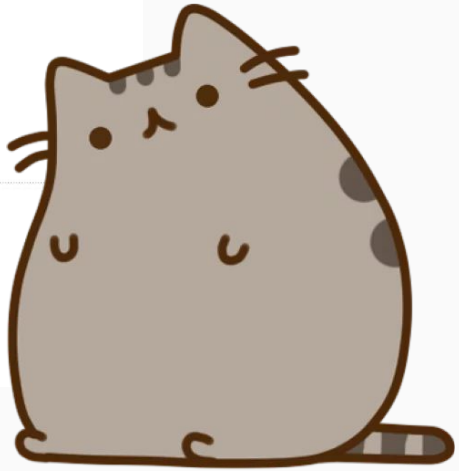
[Security Response Team] Stored Cross-Site Scripting via CLI Interface

Denis Kolegov <d.n.kolegov@gmail.com>
кому: info

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

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

Thanks.



No response ;-(

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

The Good Old Friend CSRF

CSRF Intro

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

CSRF in SD-WAN

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

SilverPeak REST API CSRF

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

Another Friend:
Host Header Attack

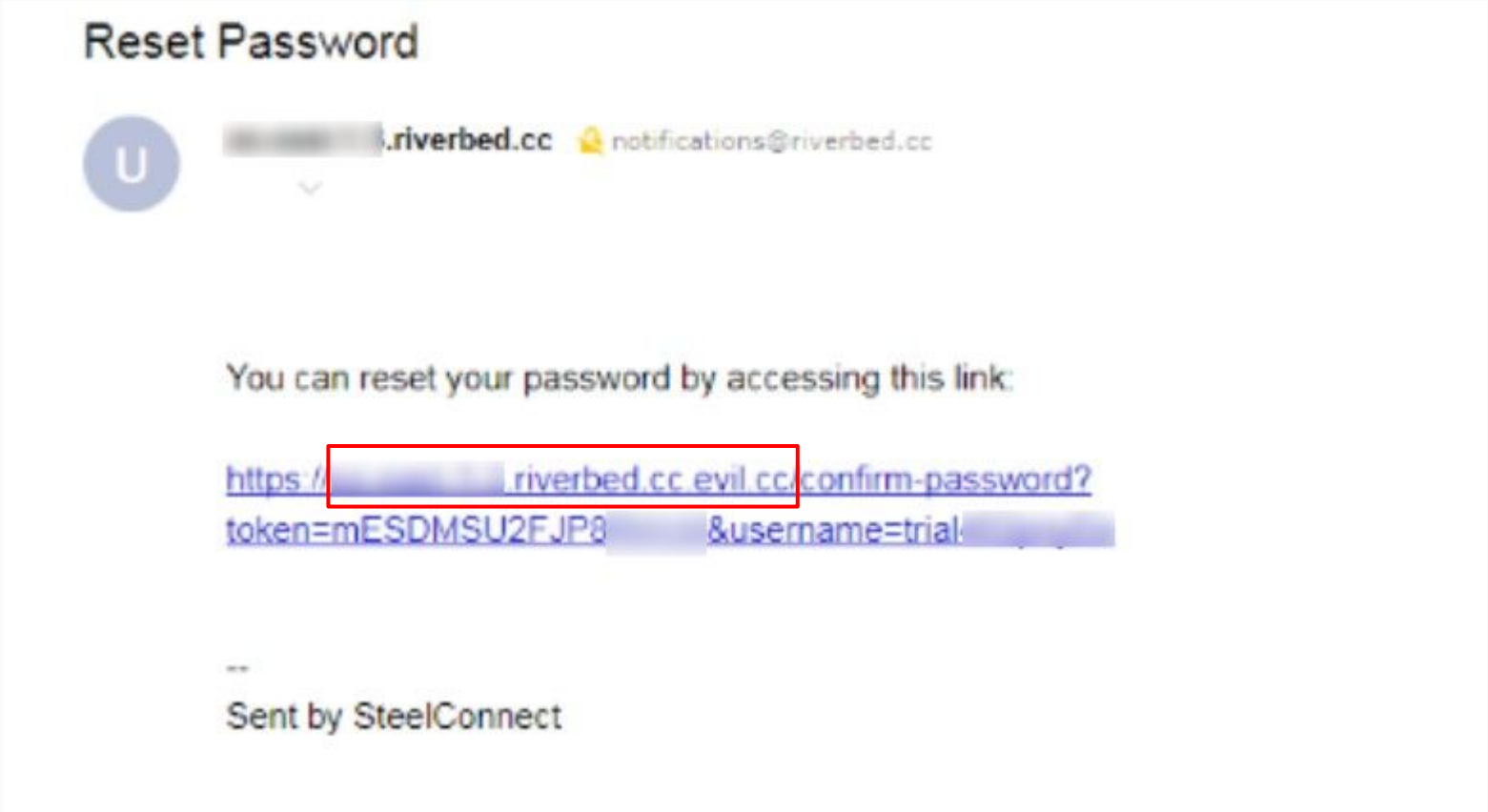
Host Header Attacks

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

Password Reset Poisoning

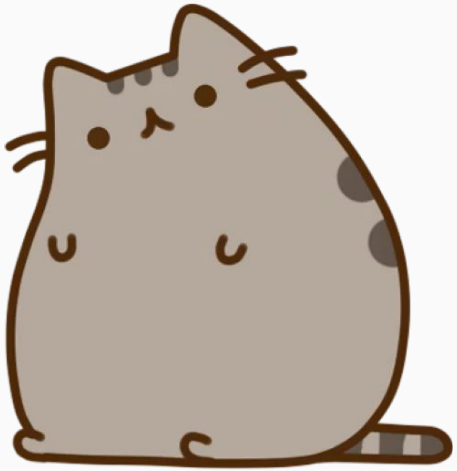
```
POST /reset-password HTTP/1.1
Host: [redacted] riverbed.cc.evil.cc
Connection: close
Content-Length: 47
Cache-Control: max-age=0
Origin: https://[redacted].riverbed.cc
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) 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



Responsible Disclosure Results

No response ;-(
Full disclosure:
https://seclists.org/fulldisclosure/2018/Oct/39



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 **did not distinguish JavaScript from NodeJS**

Client-side Authentication

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

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

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

        };

    }
```

?

!

// TODO: fix in prod ?

ZTD 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'
                        }
```

Use of Hard-coded Cryptographic Certificate

Overview

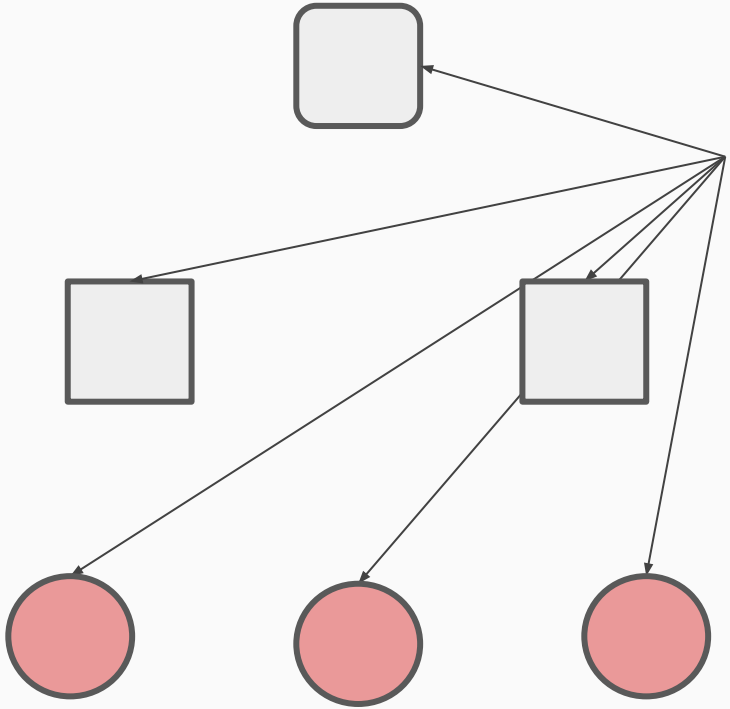
- A use of hard-coded cryptographic key was found in a one SD-WAN product (the vulnerability is being fixed now)
- All appliances use the same pre-installed PKC key pair and the corresponding self-signed certificate
- This certificate is used in Controller - Orchestrator communication protocol within 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



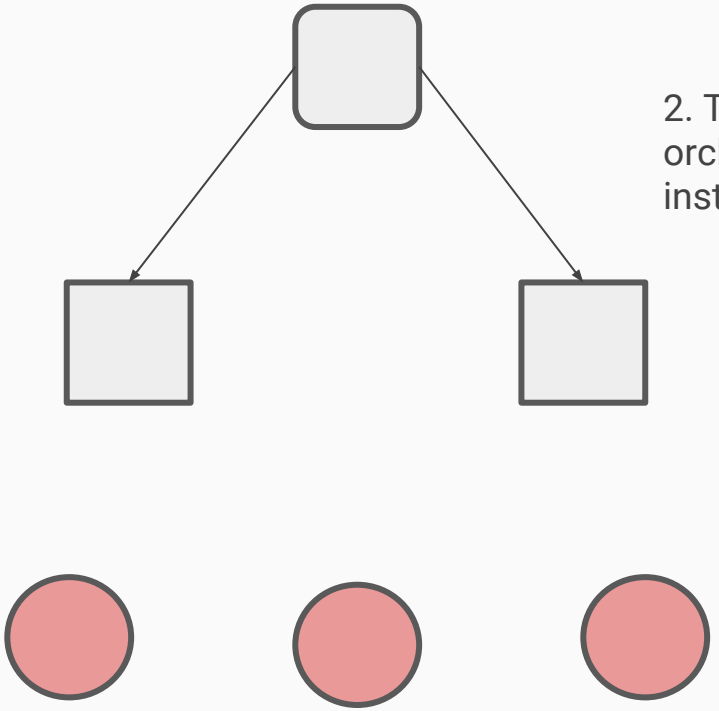
1. The Vendor copies the pre-generated appliance_cert

Provisioning (2/2)

Orchestrator

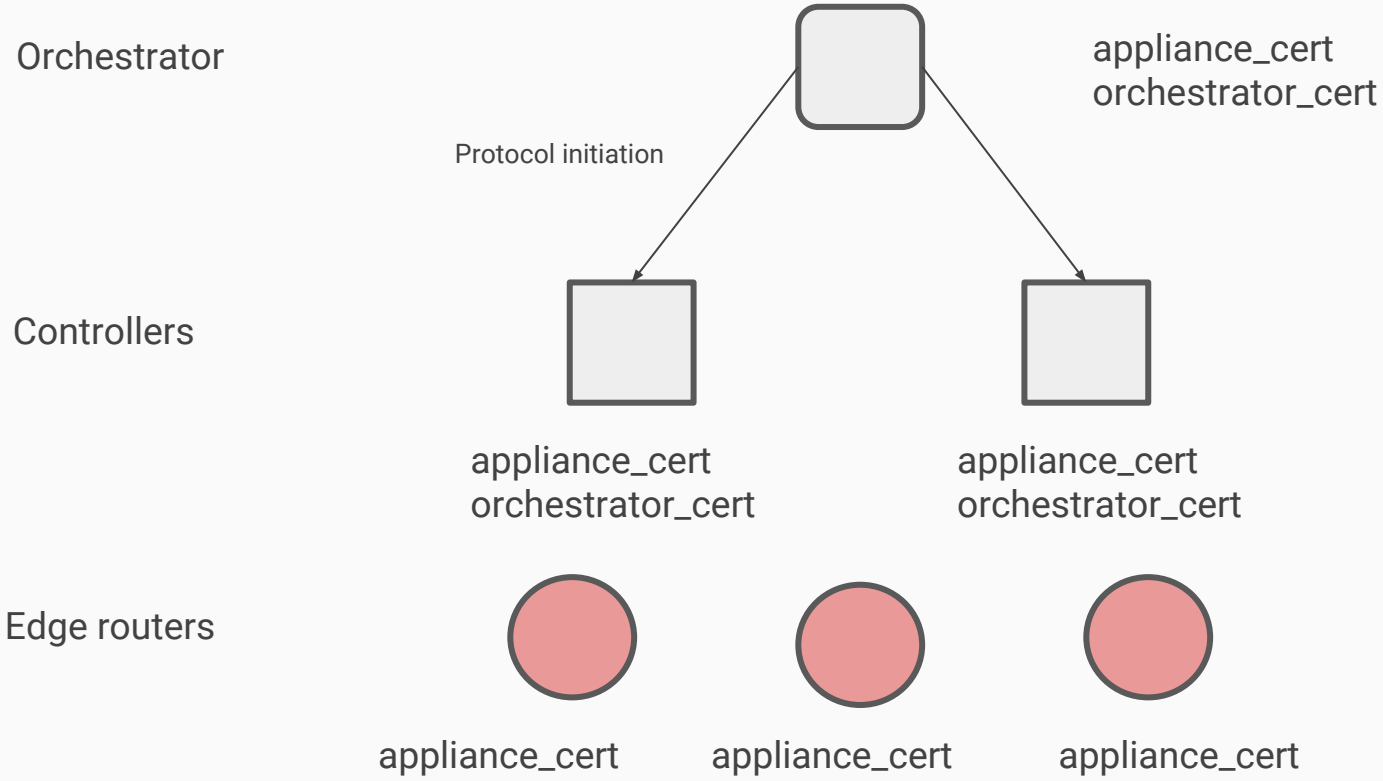
Controllers

Edge routers

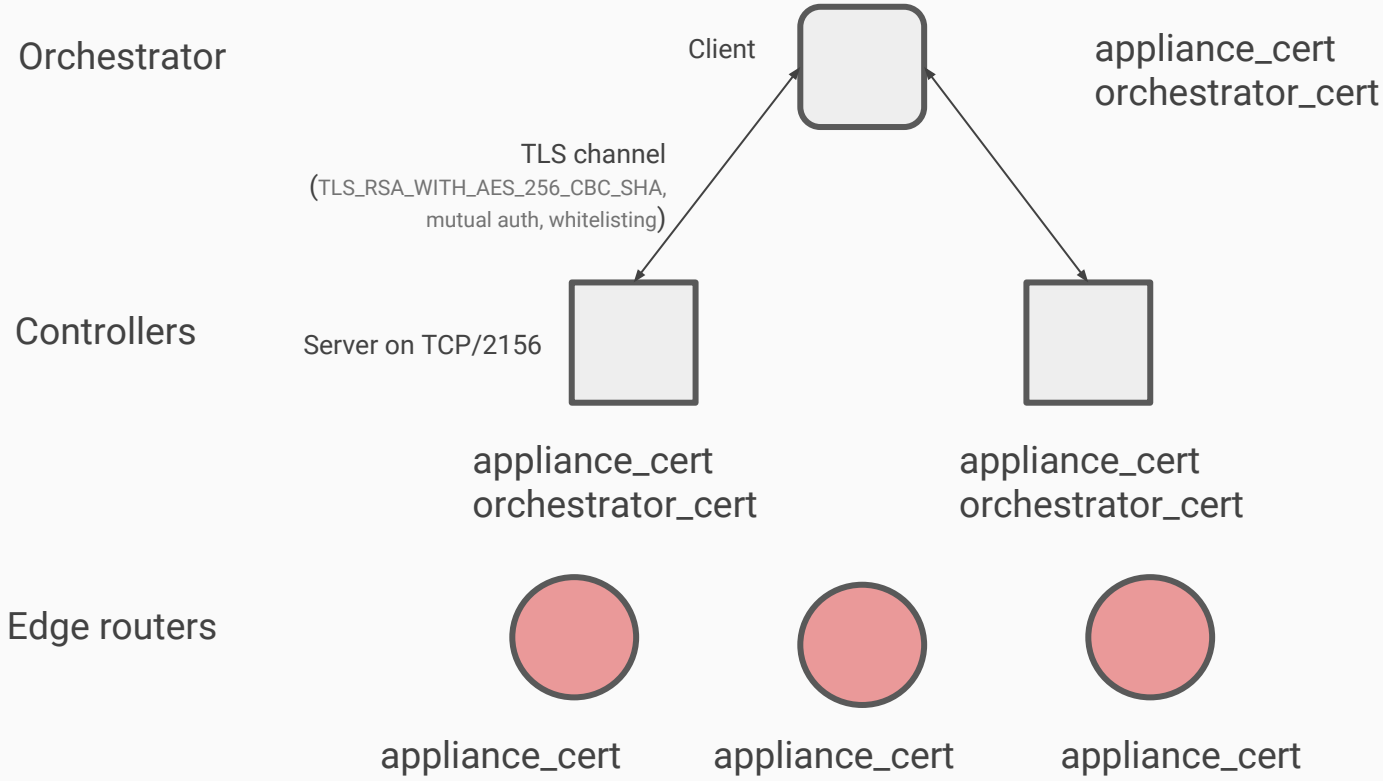


2. The Customer generates the orchestrator_cert and manually installs it on the controller nodes

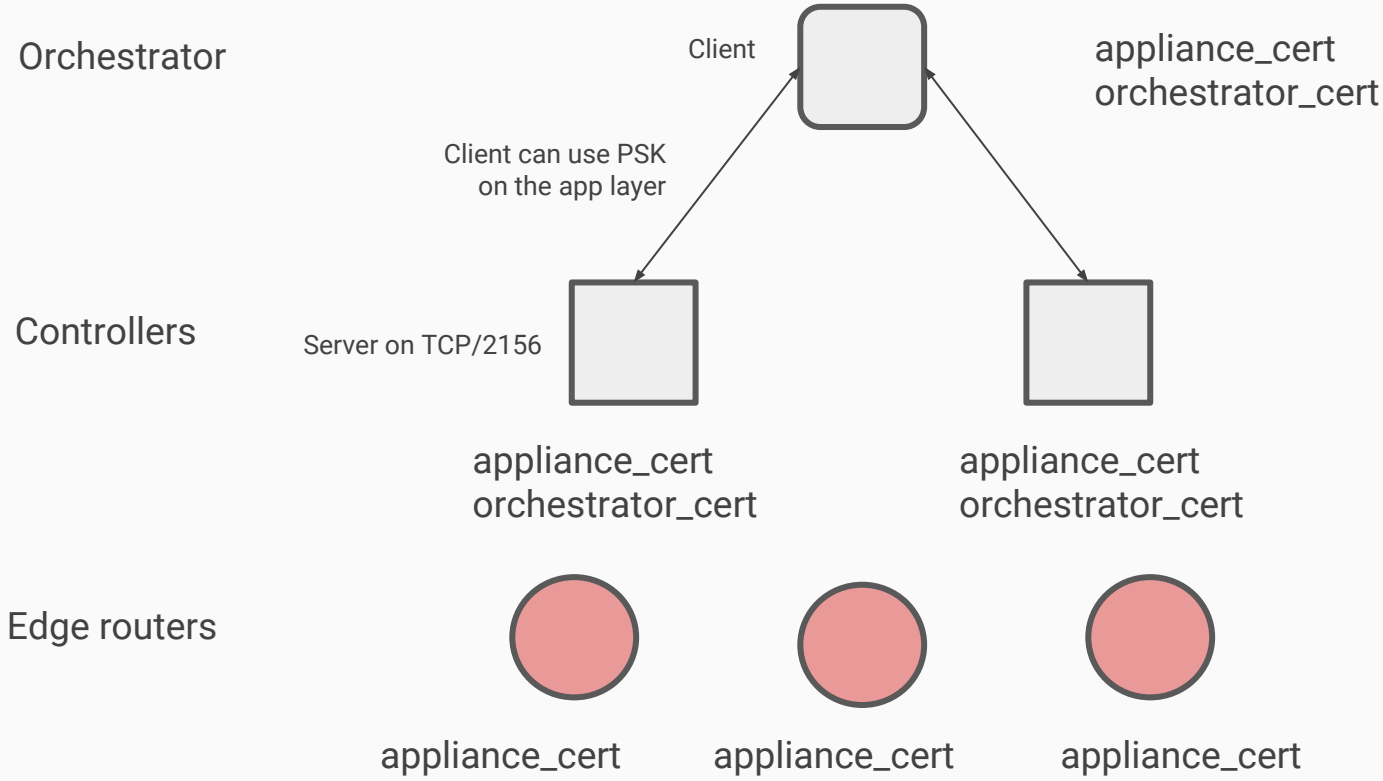
Communication Scheme (1/3)



Communication Scheme (2/3)



Communication Scheme (3/3)



Design Summary

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

appliance_cert.pem

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

TCP 2156

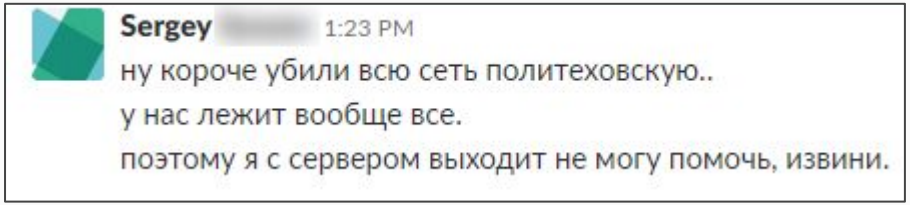
- SSL with hardcoded certificate on 2156/tcp
- Need to fingerprint SSL certificates on uncommon port
  - Shodan gives no results
  - Masscan can detect SSL and grab its certificate
- Implements a “vulncheck” function for grabbed SSL cert
- …
- EZ WIN

https://github.com/sdnewhop/masscan

Networks were harmed making the research

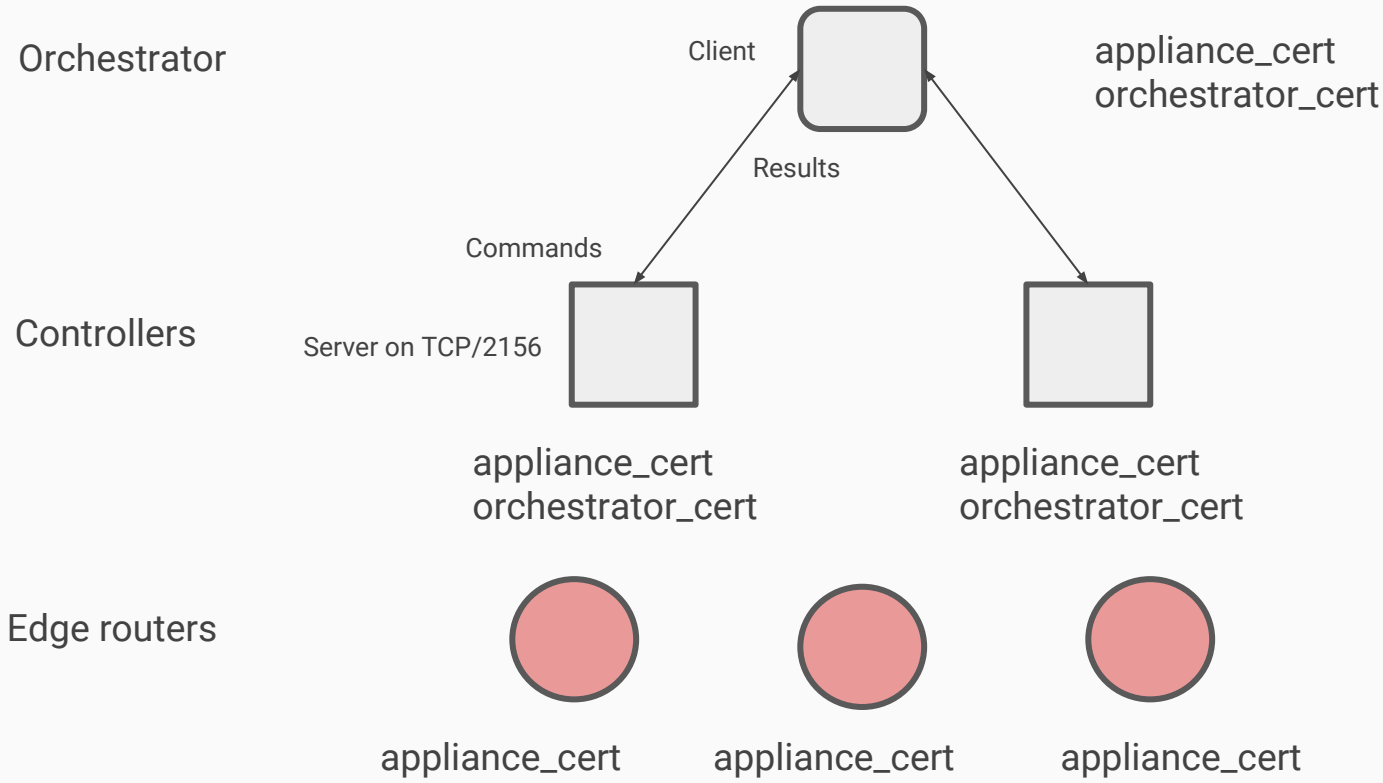
No kangaroos were harmed during research

But a network was...



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

Northbound API



What is the API and 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`)

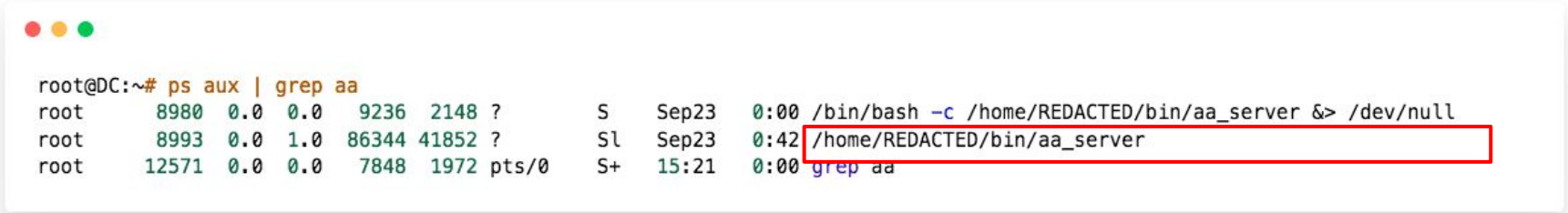
Authentication

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

Client CLI Help

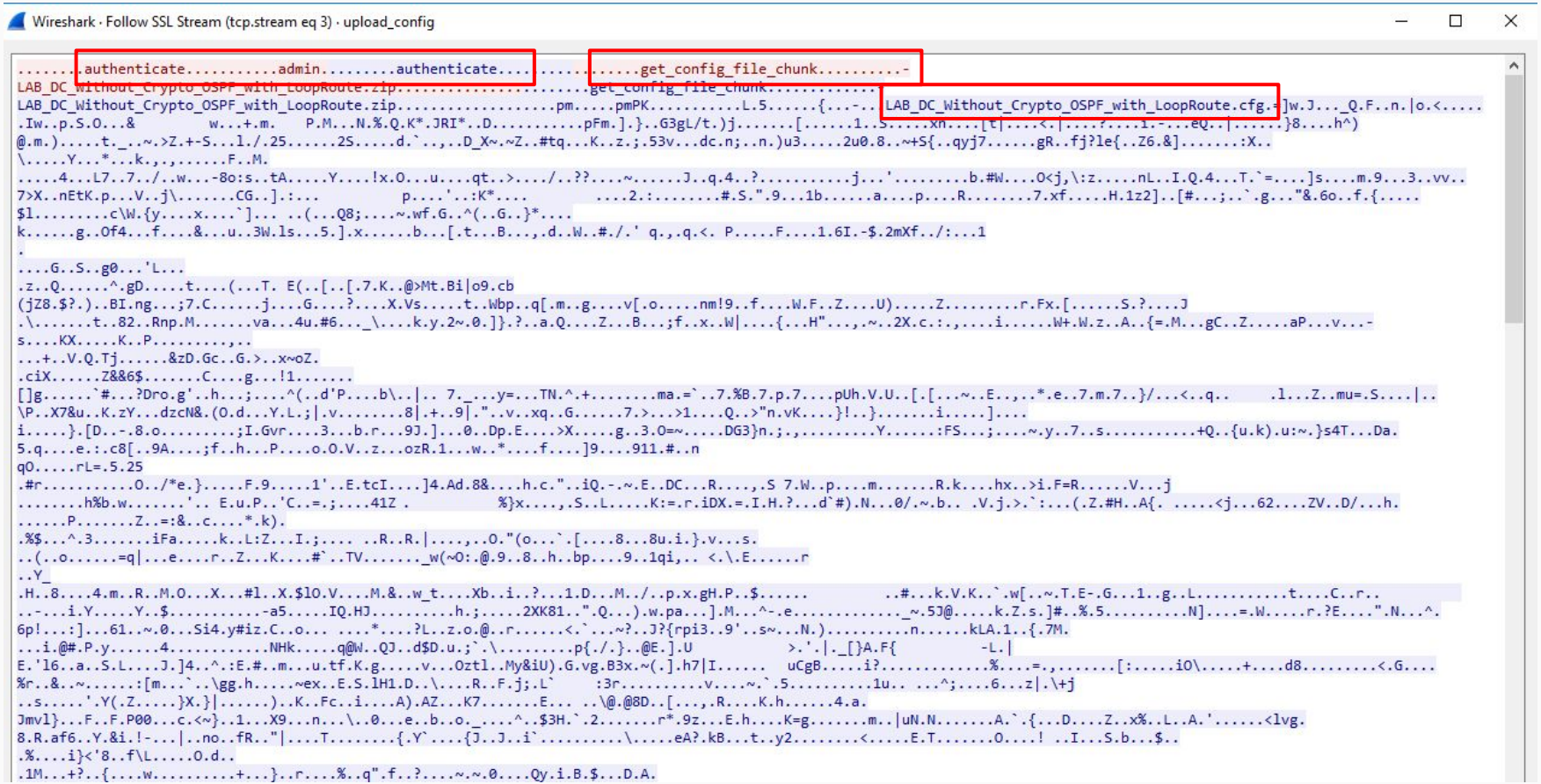
```
root@VWC:~# /home/REDACTED/bin/aa_client --help
aa_client options:
  -h [ --help ]                         print help text
  -i [ --ip_addr ] arg                  ip address of the server
  --tcp_port arg (=2156)                tcp port of the server
  -u [ --username ] arg (=REDACTED)     user name 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



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

Get Config Command with Empty Password



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 integration with a private Customer PKI
- “This hockey we do not need” (Nikolai Ozerov)

Attacker Capabilities

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**

How easy to upload a malicious certificate?

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

Responsible Disclosure Results

1. September 24, 2018: Reported
2. September 25, 2018: A bug 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”

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)
                {
                  #...
```

SNMP Route Learning

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

SNMP 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
  - No route authentication and integrity
- An attacker in MitM-position can arbitrary change routing information
- Probable RCE and SQLi in the mechanism implementation

SQLi-driven Bandwidth Detection

Automatic Bandwidth Detection

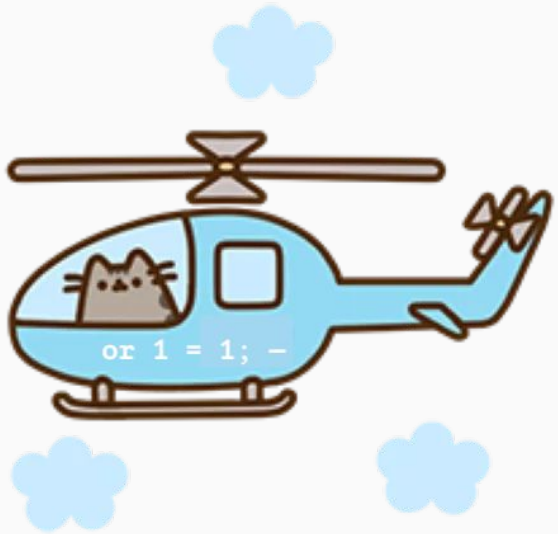
- Citrix NetScaler SD-WAN has a bandwidth-detection mechanism automatically updating the running configuration and connection policies of data plane
- The bandwidth detection feature can be scheduled to run as frequently as every hour and maintains an historical table of what the bandwidth test results were
- The current bandwidth values are stored in MariaDB
- The basic idea: If we can change bandwidth data stored in the database, we can change data plane characteristics

Automatic Bandwidth Detection

```
MariaDB [T2_Bandwidth]> describe WAN_Link_Bandwidth;
+---------------------------+---------------------+------+-----+---------+-------+
| Field                     | Type                | Null | Key | Default | Extra |
+---------------------------+---------------------+------+-----+---------+-------+
| ID                        | bigint(20) unsigned | NO   | PRI | 0       |       |
| Update_Epoch_Time_mS      | bigint(20) unsigned | NO   | PRI | 0       |       |
| Name                      | varchar(100)        | YES  |     | NULL    |       |
| WAN_Ingress_Permitted_kbps | int(10) unsigned    | YES  |     | NULL    |       |
| WAN_Egress_Permitted_kbps | int(10) unsigned    | YES  |     | NULL    |       |
+---------------------------+---------------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

Is that System vulnerable to SQLi?

- Log_monitoring_utils.cgi is vulnerable to SQLi
- Events_download.cgi is vulnerable to SQLi



SQL Injection in events_download.cgi

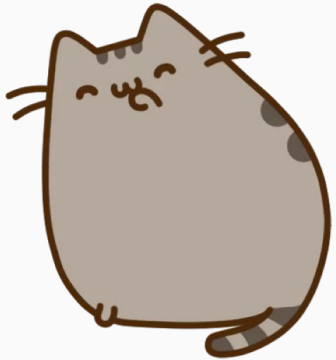
```
POST /[redacted]/events_download.cgi HTTP/1.1
Host: 10.30.37.77
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: 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

Does that system have other vulns?

Results

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



Responsible Disclosure Results

1. June 14, 2018: Reported
2. June 15, 2018: A bug created
3. October 12, 2018: A vendor have addressed reported issues and have a bulletin drafted for release. CVEs are allocated and reserved
4. October 22, 2018: the vulnerabilities were fixed
5. Citrix NetScaler SD-WAN security testing report (POC2018 special release)

Denial of Service RegEx

ReDoS in IDS Rules

```
alert http $HOME_NET any -> $EXTERNAL_NET any (
    msg:"ETPRO TROJAN Win32/Dofoil.R Checkin";
    flow:established,to_server;
    content:"POST"; nocase; http_method;
    urilen:15; content:"/site/index.php"; fast_pattern:only; http_uri;
    content:"User-Agent|3a| "; depth:12; http_header;
    pcre:"/^[a-z0-9]+?\x2b([a-z0-9]+?[\x2b\x2f\x0d\x0a]*?)+?\x3d(\x3d\x0d\x0a)?$/Pi";
    reference:md5,cf338568070f2216c9caab4d11e21be2;
    classtype:trojan-activity;
    sid:2805659;
    rev:2;
)
```

ReDoS Example

```
>>> setup = """
... import re
... def foo(m):
...     my_re = r'^[a-z0-9]+?\x2b([a-z0-9]+?[\x2b\x2f\x0d\x0a]*?)+?\x3d(\x3d\x0d\x0a)?$'
...     retrun re.findall(my_re, m)
... """
>>> timeit.timeit('foo("a+aaaa0000#a+a+=")', setup=setup, number=1)
5.0067901611328125e-05
>>> timeit.timeit('foo("a+aaaaaa000000000#a+a+=")', setup=setup, number=1)
0.003350973129272461
>>> timeit.timeit('foo("a+aaaaaaaaaaa000000000000#a+a+=")', setup=setup, number=1)
0.6693778038024902
>>> timeit.timeit('foo("a+aaaaaaaaaaaaaaa0000000000000000#a+a+=")', setup=setup, number=1)
353.6542570590973
```

Found Vulnerabilities to ReDoS

File	SID	RegExp
trojan.rules	2805659	/^[a-z0-9]+?\x2b([a-z0-9]+?[\x2b\x2f\x0d\x0a]*?)+?\x3d(\x3d\x0d\x0a)?$/Pi
trojan.rules	2805660	/^[a-f0-9]{16,20}\x3d([a-z0-9]+?(\x25(2[abf]|0d|0a))*)+?\x253d\x253d\x250d\x250a$/Pi
trojan.rules	2805643	/^\/1\.php\?id=.+?(&id=.+?)+?((&id=)?&id=)?$/Usi
web_client.rules	2805691	/(\/\/.+?\r\n|\/*.+?*\/)*?[\r\n\s]*?import\s+?[A-Z][.a-z]+?\s*?\x3b.+?[\r\n]+?function FindProxyForURL\x28url,\s*?host\s*?\x29/si
web_client.rules	2805321	/^(?P<oredirect>(\s*\d+)+)[^\]]*?\](?!endobj).)*?endobj.*?(?P=oredirect)\s*?obj[\r\n\s]*<<((?!endobj).)*?\/Subtype\s*?\/Widget((?!endobj).)+?\/FT\s*\/(?!(Btn|Tx|Ch|Sig))((?!endobj).)*?endobj/Rs
web_client.rules	2805679	/<fieldset[\r\s\n]+?([^>]+[\r\n\s]+?)*id[\r\s\n]*?\x3d[\r\s\n]*?[\x27\x22](?P<fieldsetid>[^\x22\x27]+)[\x27\x22]((?!<\/fieldset).+?<button[\r\s\n]+?([^>]+[\r\n\s]+?)*id[\r\s\n]*?\x3d[\r\s\n]*?[\x27\x22](?P<buttonid>[^\x22\x27]+)[\x27\x22]).+?<script.+?document\.getElementById\x28\s*[\x27\x22](?P=buttonid)[\x27\x22]\s*\x29\.((?!<\/script).+?(?P=fieldsetid)\.innerHTML[\r\n\s]*?\x3d[\r\n\s]*?[\x22\x27].+?CollectGarbage\x28)((?!<\/fieldset).+?)<button[\r\s\n]+?([^>]+[\r\n\s]+?)*id[\r\s\n]*?\x3d[\r\s\n]*?[\x27\x22].+?<\/fieldset>/si
web_client.rules	2805717	/var[\r\n\s]+(?P<var1>[^\r\n\s\x3d]+)[\r\n\s]*\x3d\s*?document\.getElementById\x28\s*?[\x22\x27](?P<tableid>[^\x22\x27]+)[\x27\x22]\s*?\x29.+?(?P=var1)\.(?:b(?:(?:gcolo|orde)r|ackground)|cell(?:padd|spac)ing|summary|height|align|frame|rules|width)[\r\n\s]*=.+?(?P=var1)\s*?\x3d\s*?null\s*?\x3b/si"; pcre:"/(?P<thid>[^\x2e]+)\.innerHTML[\r\s\n]*?\x3d[\r\s\n]*?[\x27\x22][\x27\x22].+?<table[\r\s\n]+?([^>]+[\r\n\s]+?)*id[\r\s\n]*?\x3d[\r\s\n]*?[\x27\x22](?P<tableid>[^\x22\x27]+)[\x27\x22]((?!<\/table).+?<th[\r\s\n]+?([^>]+[\r\n\s]+?)*id[\r\s\n]*?\x3d[\r\s\n]*?[\x27\x22](?P=thid)[\x27\x22])((?!<\/th).+?)<(?:s(?:trong>.*?<\/strong|amp>.*?<\/samp)|code>.*?<\/code|dfn>.*?<\/dfn|kbd>.*?<\/kbd|var>.*?<\/var|em>.*?<\/em)>.+?<\/th>.+?<\/table>/s
web_client.rules	2017479	/^[\r\n\s]+(?P<func>[^\r\n\s]+)[\r\n\s]*?\([^\)]*?\)[\r\n\s]*?\{((?!function).)*?(\b(?P<var>[^\r\n\s\=]+)[\r\n\s]*?=[\r\n\s]*?(?:\x22\x22|\x27\x27))?((?!function).)*?document\.write\([\r\n\s]*?(?:\x22\x22|\x27\x27|(?P=var))[\r\n\s]*?\).+?onlosecapture(:?([\x22\x27][\r\n\s]*?\])?[\r\n\s]*?=|[\x22\x27][\r\n\s]*?\,)[\r\n\s]*?(?P=func)\b/Rsi
web_server.rules	2002997	/\.php.+(path|page|lib|dir|file|root|icon|lang(uage)?|folder|type|agenda|gallery|domain|calendar|settings|news|name|auth|prog|config|cfg|incl|ext|fad|mod|sbp|rf|id|df)[a-z](\[.*\])+)\s*=\s*https?/Ui
telnet.rules	2800058	/\x03(OS|Path|SystemRoot|WinDir|HOMEDRIVE|USERNAME|USERDOMAIN)((\x00|\x01|\0x02|\0x03).*)*\xFF\xF0$/RBi
current_events.rules	2018171	/^\W/R"; within:100; content:"if"; distance:-200; within:200; nocase; pcre:"/^(?:\s*?|\/*(?:(?!*\/).)*?*\/)*?\((?:\s*?|\/*(?:(?!*\/).)*?*\/)*?(?P<vname>[^\s>=]+)(?:\s*?|\/*(?:(?!*\/).)*?*\/)*?<(?:\s*?|\/*(?:(?!*\/).)*?*\/)*?32\b.{0,200}(?P=vname)(?:\s*?|\/*(?:(?!*\/).)*?*\/)*?\x3d(?:\s*?|\/*(?:(?!*\/).)*?*\/)*?63\b.{1,200}\+=.{0,200}\((?:\s*?|\/*(?:(?!*\/).)*?*\/)*?(?P=vname)/Rsi
exploit.rules	2800370	/^([^\x2c\x0a]+\x2c)*\s*[^\x3d\x3b\x2c\x0a]{37}/R

Are Orchestrators More Secure?

Not so Secure :(

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

Command Injection

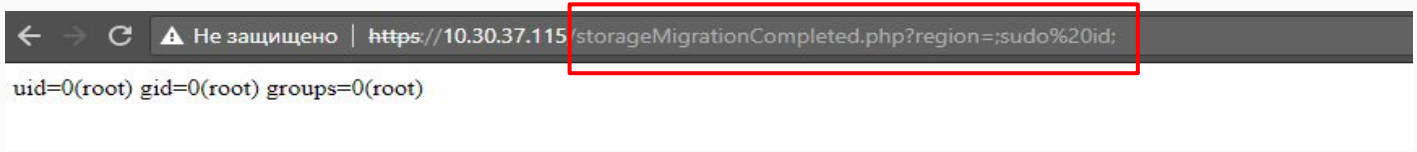
- The vulnerability in "`/app/webroot/storageMigrationCompleted.php`" leads to OS command injection attack
- An attacker without any privileges can perform this attack
- It must have a network connection to the Web Management Interface only

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");
```



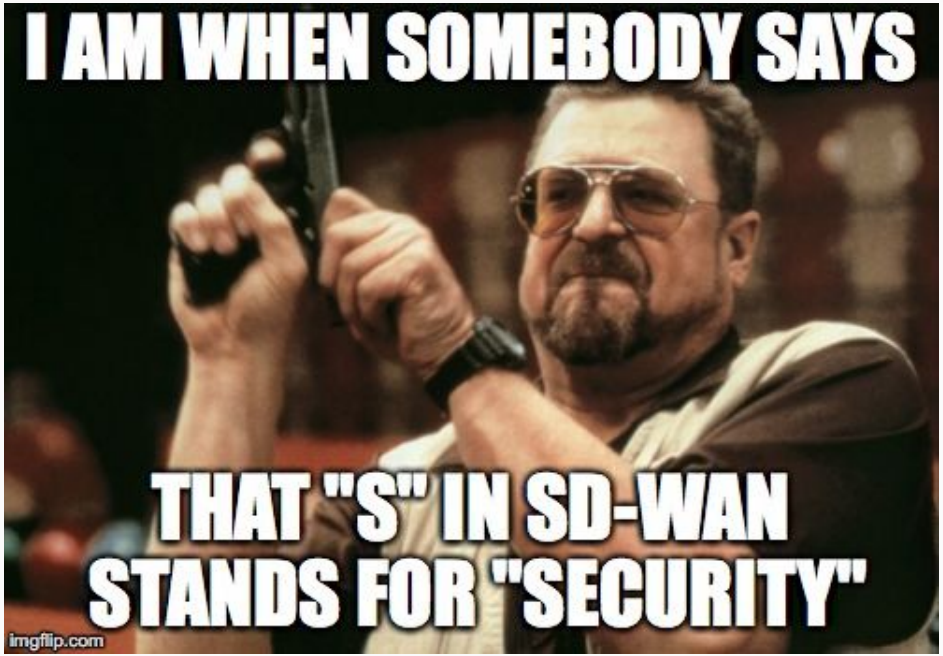
Responsible Disclosure Results

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

Conclusions

Conclusions

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



Any Questions?

Thanks!

Contact us:

@dnkolegov
@yalegko