Why Nation-State Malwares Target Telco Networks: Dissecting Technical Capabilities of Regin and Its Counterparts

Author: Ömer Coşkun

The supreme art of war is to subdue the enemy without fighting. Sun Tzu
Outline

- Overview
  - Telecom Network Architecture
  - Practical Attack Surfaces
    - GRX Attack Vectors
    - SS7 Attack Vectors
    - Practical Attack Scenarios

- Rootkit Attacks: Regin and it’s counterparts
  - Common Rootkit Techniques and Regin
  - Regin vs. Uruborus and Duqu
  - Demo: PoC || GTFO

- Questions?
Ömer Coşkun (@0xM3R)

- BEng. Computer Science
- Research Assistant in Quantum Cryptography & Advanced Topics in AI

- Industry Experience
- **KPN** – CISO, Ethical Hacking
- **Verizon** – Threat & Vulnerability Management
- **IBM ISS** – Threat Intelligence

- Interests
- Algorithm Design, Programming, Cryptography, Reverse Engineering, Malware Analysis, OS Internals, Rootkits
Motivations

- Analyze existing vulnerabilities and attack surface of GSM networks
- Governments hack their own citizens
- Surveillance implants shifted focus to telecom networks and network devices
- European Telco companies are really paranoid after Regin attack
- Rootkits are fun: a lot to learn & challenge
- Reproduce the attack scenario and implement it!
GSM Network Architecture
GSM Network Architecture

- Traffic (voice 13kbps, data 9.6kbps)
- Signaling LAPDm
Regin targets GSM Networks

Regin: nation-state ownage of GSM networks
"Beware of Regin, the master! His heart is poisoned. He would be thy bane..."

By GReAT on November 24, 2014, 2:00 pm

'Regin': The 'New Stuxnet' spook-grade
SOFTWARE WEAPON described
'A degree of technical competence rarely seen'

Regin Is ‘Groundbreaking’
Malware Used by UK Spooks

November 24, 2014 // 01:16 PM EST
Determining Attack Surface

G.C.H.Q.
ALWAYS LISTENING TO OUR CUSTOMERS

The NSA
The only part of government that actually listens.

WITH THIS TECHNOLOGY
WE WILL BRING THE UNITED STATES TO ITS KNEES

Script Kiddies
Cyber Criminals
APTs

Attack Surface

TECHNICAL SOPHISTICATION
Determining Attack Surface
Potential Attack Surfaces

- Absence of physical intrusion detection devices
- Vulnerable services running accessible from BTS
- Absence of tamper resistance and unauthorized access protection
- Improper network segmentation; inner non-routable segments of the Telco company could accessible.
- Core GPRS Network and Network Subsystem (NSS) could be exploitable!
Potential Attack Surfaces
GRX Networks

Mobile IP traffic:
- GPRS
- MMS eXchange (MMX)
- SMS eXchange (SMX)

Roaming mobile data user

Home mobile operator

Roaming mobile data user

Roaming mobile data user
GRX Networks

- GPRS roaming exchange, interconnecting networks.
- Your local GSM provider abroad
- Trust-based, highly interconnected network, made for internet sharing
- A failure or malicious activity would affect multiple connected machines
- Multiple attacks vectors, not limited to a particular segment where you are originating from.
- GPRS roaming exchange, interconnecting networks.

- Your local GSM provider abroad

- Trust-based, highly interconnected network, made for internet sharing

- Multiple attacks vectors, not limited to a particular segment where you are originating from.
Juicy information is here.

GRX Networks – Network Flow
And more juicy information is here.
GRX Networks – Attacks & Flaws

Are you telling me all your communication intercepted and logged including your physical location?

NSA: We'll move your metadata into /dev/null if you stop suing us

The NSA promises to delete its phone metadata early next year

Digital Trends - 2 days ago

The National Security Agency will lose the power to keep and track phone metadata

"WE KILL PEOPLE BASED ON META-DATA"
SS7 & SIGTRAN

SS7 Stack
- Radio interface related
- BSSAP
- MAP
- CAP
- INAP
- TCAP
- SCCP
- MTP3
- MTP2
- MTP1/Hardware

SIGTRAN Stack
- ISUP
- TUP
- TCAP
- SCCP
- MTP3
- M2PA
- M2UA
- M3UA

Implementation:
- Green: Implemented
- Yellow: Roadmap - Inprogress
- Black: Will not implement

http://www.telestax.com/
SS7 Introduces procedures for
- User identification.
- Routing
- Billing
- Call management
SS7 Features:

- Flow control of transmitted information
- Traffic congestion controls
- Peer entity status detection (GT + PC or SPC)
- Traffic Monitoring and monitoring measurements
SS7 & SIGTRAN

SS7 over IP (SIGTRAN) Analysis and Simulation
SS7 & SIGTRAN
SS7 Protocol Analysis

### SS7 Protocol Analysis SS7 ITU

<table>
<thead>
<tr>
<th>Dev</th>
<th>TS...</th>
<th>Su...</th>
<th>Frame#</th>
<th>TIME (Relative)</th>
<th>Len</th>
<th>BSN</th>
<th>BIB</th>
<th>FSN</th>
<th>FIB</th>
<th>Sta...</th>
<th>SLC</th>
<th>DPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>00:00:00.000000</td>
<td>21</td>
<td>34</td>
<td>1</td>
<td>68</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2.22</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>1</td>
<td>1</td>
<td>00:00:00.120500</td>
<td>21</td>
<td>34</td>
<td>1</td>
<td>69</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2.22</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>2</td>
<td>2</td>
<td>00:00:00.133750</td>
<td>21</td>
<td>34</td>
<td>1</td>
<td>70</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2.22</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>3</td>
<td>3</td>
<td>00:00:00.216000</td>
<td>21</td>
<td>34</td>
<td>1</td>
<td>71</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2.22</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>4</td>
<td>4</td>
<td>00:00:00.281500</td>
<td>21</td>
<td>34</td>
<td>1</td>
<td>72</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2.22</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>5</td>
<td>5</td>
<td>00:00:00.361500</td>
<td>21</td>
<td>34</td>
<td>1</td>
<td>73</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2.22</td>
</tr>
</tbody>
</table>

Card 1 TimeSlot=20 Frame=0 at 00:00:00.000000 OK Len=21

HDLC Frame Data + FCS

```
====== MTP2 Layer ======
BSN = .0100010 (34)
BIB = 1........ (1)
FSN = .1000100 (68)
FIB = 1........ (1)
LI = .010011 MSU Format

====== MTP3 Layer ======
Service Indicator
Priority Code
Sub-service field
DPC
OPC
Signalling Link Code

====== ISUP Layer ======
Circuit Identification Code
```

Hex Dump of the Frame Data

```
A2 C4 13 05 12 50 02 02 01 00 0C 02 00 05 C2 10 
00 00 0B C7 F8
```

Off-line Viewing

D:\Program Files\GL Communication\2349 Frames
SS7 Protocol Analysis

All the juicy info here:

- Calling no.
- Called no
- Call duration
- Call status

![SS7 Protocol Analysis S7 ITU](image)
Feel confident that NSA not interested in ‘Good’ people?

SS7 Protocol Attacks & Flows

**NSA: We'll move your metadata into /dev/null when you stop suing us**

November 29 set as cutoff date for further collection

NSA Not Interested in “Bad People”; It Listens to “Interesting ...”

www.youtube.com/watch?v=BhrPVse9TnE

Feb 4, 2015 - Uploaded by Ken Campbell

Michael Hayden, former NSA Director, former Director of National Intelligence, former Director of CIA has ...

"WE KILL PEOPLE BASED ON META-DATA"
SS7 Practical Attack Scenarios

• Intercepting subscribers calls
• Subscriber service change attacks
SS7 Practical Attack Scenarios

- Interception of SMS messages
- Interception of outgoing calls
- Redirection of incoming or outgoing calls
- Making changes in user bills or balance
Unblocking stolen mobile devices using SS7-MAP vulnerabilities

Exploiting the relationship between IMEI and IMSI for EIR access

Siddharth Prakash Rao¹,², Silke Holtmanns¹, Ian Oliver¹, Tuomas Aura²
¹Nokia Networks
Espoo, Finland
{sid.rao, silke.holtmanns, ian.oliver}@nokia.com
²Department of Computer Science and Engineering
Aalto University, School of Science
Espoo, Finland
{siddharth.rao, tuomas.aura}@aalto.fi

Abstract— The increase in usage of mobile phones and the relative increase in the number of mobile phone thefts have imposed an overhead on securely retrieving the stolen or missing devices. While the mobile security researchers try to figure out various mechanisms to track such devices, attackers on the other hand are trying to exploit weaknesses in the mobile network system to dissipate into the dark side with stolen devices. In this paper we present how the SS7-MAP protocol can be misused to help an attacker to unblock the device from the stolen list and use it normally.

Keywords— SS7, MAP, EIR, Stolen Mobiles, IMEI

over stolen/missing devices and the private information on them. There exists a growing black market [3] where such devices are sold with least possibility of tracing them.

Signaling System No.7 (SS7) is a mobile backend protocol used for interconnectivity between mobile operator networks which enables roaming and cellular services across operator domains. This protocol is mainly used for communication between the network elements and between networks. Recent successful attacks on SS7 as per [4][5][6][7] and [8] haven proven that an attacker with privilege over the core mobile network can take control over users personal information such as billing data and Short Message Service (SMS) messages.
SS7 Practical Attack Scenarios

- Unblocking stolen mobile devices

1. Attacker turns on the “IMSI Check” option if EIR.

2. MSC sends the modified MAP_CHECK_IMEI.

3. Parse the message to derive IMEI and IMSI.

4. Check the IMEI obtained from step 3, black, white and grey lists.

5. If that IMEI is found in blacklist, use the IMEI-IMSI pair obtained in step 3 in the EIR entry of IMEI-IMSI pairs.

6. If a match is found in step 5, override the blacklist and move that IMEI to whitelist.

7. Send MAP_CHECK_IMEI_ACK message to allow registration further.
On 5 Sep, 2013, at 3:10 PM, Alberto Ornaghi <a.ornaghi@hackingteam.com> wrote:

I think that any telco in the world could access SS7 signals to know where a phone number is located. If the have to call that number from a country, let’s say UK, the UK’s provider must know where the phone is attached to the global network since if the called one is in UK the call is not routed abroad and routed directly internally. Sounds plausible?

On Sep 5, 2013, at 09:03, serge <s.woon@hackingteam.com> wrote:

To narrow down the possible whistleblower, I think the person has to be working in the telecom provider we subscribed to ... TIM? Location query from SS7 signalling can only indicate the location of the mobile phone at the time of query (we can verify this with NICE :-)). The records are clearly a database query as it is very systematic; twice per day at 1100h and 2300h. As the records started from one or two years back, either Mus and Marco.B have a common secret admirer (which I do not think so) or the person just query the records from the telco existing database knowing that their names are mentioned in controversial articles. We can compliant to TIM (I am not sure whether they will do anything about it) or just switch to another telco and monitor the situation.

Source: https://wikileaks.org/hackingteam/emails/emailid/343623
Rootkit Techniques

- Exploitation/Dropper
- USER APPS
  - EXE
  - DLL
- WINAPI
- User Mode
- Kernel Mode
- HAL
- IAT Hooking
- SSDT Hooking
- IDT Modification
- IRP Dispatch Table
- Hooking through CPU MSR
**Captain Hook Style Hacking:** Intercepts every function, keeps a copy of the content for herself, and then let the function continue as it was supposed to ...
Rootkit Techniques

User Mode

Kernel Mode

Rootkit Techniques

DLL Injections
IAT Hooking
Inline Hooking

User Mode

SSDT Hooking
IRP Hooking
IDT Hooking

Kernel Mode

GDT Hooking
SYSENCER Hooking
Regin Platform Analysis

Challenges, Hurdles & Difficulties:

- No one had the dropper when started analysis
- Multi stage and encrypted framework structure
- Modules are invoked via SOA structure by the framework
- Malware data are stored inside the VFS
- Researched GSM Networks had no indication of compromise 😊
Regin Platform Analysis

- What is the solution?

- RE Orchestrator
- Memory dumps
- Static Analysis
- Instrumentation of Calls
- Dynamic Analysis

Check similar work & the write up:
http://artemonsecurity.com/regin_analysis.pdf
Regin Platform Stages

REGIN FRAMEWORK

STAGE 1 -> LOADERS

Dropper

Stage 1

User Mode

Stage 2

Kernel Mode

Stage 3

kpn
Regin Platform – Stage 1

Stage 1

<table>
<thead>
<tr>
<th>Name</th>
<th>Virtual Size</th>
<th>Virtual Address</th>
<th>Size of Raw Data</th>
<th>Pointer to Raw Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>.text</td>
<td>00002578h</td>
<td>00011000h</td>
<td>00002600h</td>
<td>00000400h</td>
</tr>
<tr>
<td>.data</td>
<td>00000104h</td>
<td>00014000h</td>
<td>00000200h</td>
<td>000002A00h</td>
</tr>
<tr>
<td>.data</td>
<td>0000033Ch</td>
<td>00015000h</td>
<td>00000400h</td>
<td>000002C00h</td>
</tr>
<tr>
<td>INIT</td>
<td>000002E0h</td>
<td>00016000h</td>
<td>00000400h</td>
<td>00000300h</td>
</tr>
<tr>
<td>.reloc</td>
<td>00000396h</td>
<td>00017000h</td>
<td>00000400h</td>
<td>000003400h</td>
</tr>
</tbody>
</table>

Unprocessed data

jmp_ntosknl.exe+except_handler3:

jmp [ntosknl.exe+except_handler3]

SUB_0001108E:

mov edi,edi
push ebp
mov ebp,esp
sub esp,00000020h
mov eax,[ebp+0Ch]
mov byte ptr [ebp-01h],00h
Regin Platform – Stage 2

ObjectAttributes = OBJECT_ATTRIBUTES ptr -34h
DestinationString = UNICODE_STRING ptr -1Ch
ValueName = UNICODE_STRING ptr -18h
Length = dword ptr -6Ch
KeyHandle = dword ptr -8
ResultLength = dword ptr -4

mov edi, edi
push ebp
mov ebp, esp
sub esp, 34h
push esi
mov esi, ds:RtlInitUnicodeString
push edi
push offset SourceString ; "\REGISTRY\Machine\System\CurrentCon"
lea eax, [ebp+DestinationString]
xor edi, edi
push eax ; DestinationString
mov [ebp+ResultLength], edi
call esi ; RtlInitUnicodeString
push offset word_163E6 ; SourceString
lea eax, [ebp+ValueName]
push eax ; DestinationString
call esi ; RtlInitUnicodeString
lea eax, [ebp+DestinationString]
lea eax, [ebp+ObjectName]
mov [ebp+ObjectAttributes.ObjectName], eax
lea eax, [ebp+ObjectAttributes]
push eax ; ObjectAttributes
push_03Ch

RegistryPatha = RtlWriteRegistryValue(
    SourceString.Buffer, Destination.Buffer,
    ValueData,
    (unsigned __int16)v18 + 2);
if (RegistryPatha < 0 )
    break;
}
Regin Platform – Stage 2

```c
Key * KeyHandle = 0;
RtlInitUnicodeString(&DestinationString,
"\REGISTRY\Machine\System\CurrentControlSet\Control\Session Manager\Memory Management");
RtlInitUnicodeString(&ValueName, &word_163E6);
ObjectAttributes.ObjectName = &DestinationString;
ObjectAttributes.Length = 24;
ObjectAttributes.RootDirectory = 0;
ObjectAttributes.Attributes = 64;
ObjectAttributes.SecurityDescriptor = 0;
ObjectAttributes.SecurityQualityOfService = 0;
result = ZwOpenKey(&KeyHandle, 0xF003Fu, &ObjectAttributes);
if ( !result )
{
    if ( ZwQueryValueKey(KeyHandle, &ValueName, KeyValueFullInformation, 0, 0, &ResultLength) == -1073741798 )
    {
        Length = ResultLength;
        v1 = sub_12892(ResultLength, PagedPool);
        if ( v1 )
        {
            if ( !ZwQueryValueKey(KeyHandle, &ValueName, KeyValueFullInformation, (POVlD)v1, Length, &ResultLength) & (DWORD *)(v1 + 8) == 1 )
                byte_19879 = 1;
        sub_12952(PVOID)v1, Length);
memset(v5, 0, ObjectName, MaximumLength);
RtlAppendUnicodeToString(&ObjectName, L"\Device\\");
RtlAppendUnicodeToString(&ObjectName, v16.Buffer);
v14 = ExAllocatePoolWithTag(0, 4 * v12, 0x206B6444u);
if ( v14 )
{
    v22 = 0;
    for ( Value = 0; Value < v12; ++Value )
    {
        RegistryPatha = RtlIntegerToString(Value, 0xAu, &DestinationString);
        (RegPath = RtlAppendUnicodeToString());
```
ExAllocatePoolWithTag routine

The ExAllocatePoolWithTag routine allocates pool memory of the specified type and returns a pointer to the allocated block.

Syntax

C++

```c
VOID ExAllocatePoolWithTag(
    _In_ POOL_TYPE PoolType,
    _In_ SIZE_T NumberOfBytes,
    _In_ ULONG Tag
);
```

```
00029232 aNtoskrnl_exe  db 'ntoskrnl.exe',0
0002923F align 10h
00029240 unk_29240 db 3Ah ; ; DATA XREF: seg000:000291C8To
00029241 align 2
00029242 aExallocatepool db 'ExAllocatePoolWithTag',0
00029258 unk_29258 db 47h ; G ; DATA XREF: seg000:000291C4To
00029259
```
VOID KeStackAttachProcess(
    _Inout_ PRKPROCESS   Process,
    _Out_    PRKAPC_STATE ApcState
);

db 3
    22h ;
    2
    50h ;
    P
    48h, 65h
    1Eh
    2, 48h, 65h
    60h ;
    4, 5Ah, 77h
    4Fh ;
    2
    0F3h ;
    2
Regin Platform – Stage 3 & 4 – How to Weaponize it?

1. Register a call-back function to a process
2. Log the PID of the target process
3. Obtain PEB via `ZwQueryInformation()` for base addresses of the modules
4. Obtain the EP via `PsLookupProcessByProcess()`
5. Get inside to the process context via `KeStackAttachProcess()` referenced by EP
6. Read PEB and other data in process context
Regin Platform – Stage 3 & 4 – How to Weaponize it?

```c
InitializeObjectAttributes(&ObjectAttributes, NULL, OBJ_KERNEL_HANDLE, NULL, NULL);

clientID.UniqueProcess = hProcessId;
clientID.UniqueThread = NULL;

__try{

    status = ZwOpenProcess(&hProcessHandle, DesiredAccess, &ObjectAttributes, &clientID);

    status = ZwQueryprocess(hProcessHandle, ProcessBasicInformation, (PVOID)&BasicInfoReal, sizeof(PROCESS_BASIC_INFORMATION), &SizeReturned);

    status = PsLookupProcessByProcessId(hProcessId, &ep);

    __asm{
        mov eax, ep
        mov eax, [eax]
        mov myPwnProcess, eax
    }
}

KeStackAttachProcess(&myPwnProces, ka_state);
```
<table>
<thead>
<tr>
<th></th>
<th>Uruborus</th>
<th>Regin</th>
<th>Duqu2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypted VFS</td>
<td>Encrypted VFS</td>
<td>Encrypted VFS #2</td>
<td></td>
</tr>
<tr>
<td>PatchGuard Bypass</td>
<td>Fake Certificate</td>
<td></td>
<td>Stolen Certificate</td>
</tr>
<tr>
<td>Multiple Hooks</td>
<td>Orchestrator SOA</td>
<td></td>
<td>Orchestrator SOA</td>
</tr>
<tr>
<td>AES</td>
<td>RC5</td>
<td>Camellia 256, AES, XXTEA</td>
<td></td>
</tr>
<tr>
<td>Backdoor/Keylogger Mod</td>
<td>Advanced Network/File Mods</td>
<td>More Advanced Network/File/USB Mods</td>
<td></td>
</tr>
</tbody>
</table>
Mini Regin Attack Simulator

- Covert Channel Data Exfiltration
- Run as a thread of legitimate app’s address space
- Orchestrator simulator and partial SOA
- File system, registry and network calls hooking
- Backdoor/Keylogger Mod
Demo
Questions ?
Thank you very much for your attention
References

- http://4g-lte-world.blogspot.nl/2013/03/gprs-tunneling-protocol-gtp-in-lte.html
- http://www.gl.com/ss7_network.html
- http://www.slideshare.net/mhaviv/ss7-introduction-li-in