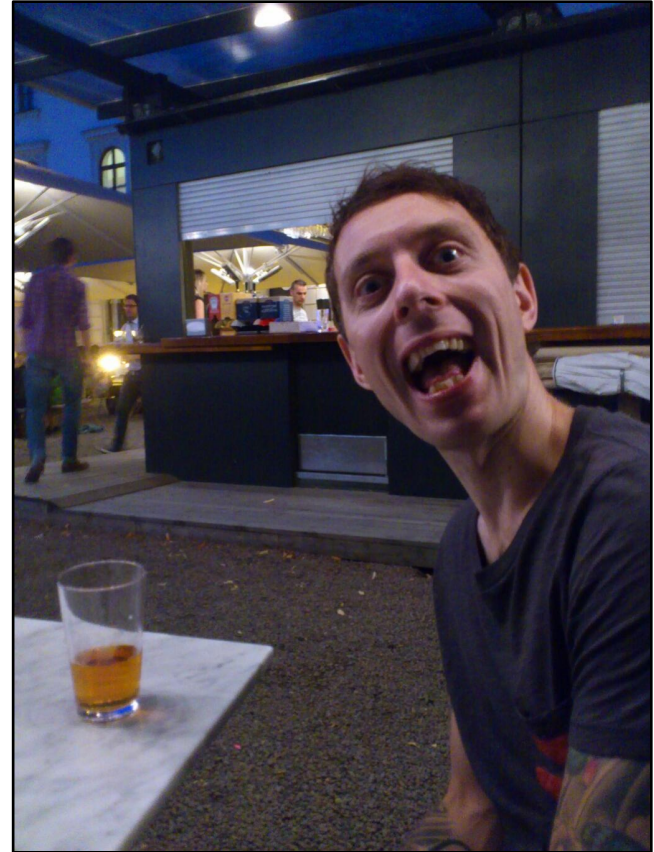


Attacking Mobile Broadband Modems Like A Criminal Would

Andreas Lindh, @addelindh, Black Hat USA 2014

whoami

- Security Analyst with I Secure Sweden
- Technical generalist
- I like web
- Not really an *expert* on anything

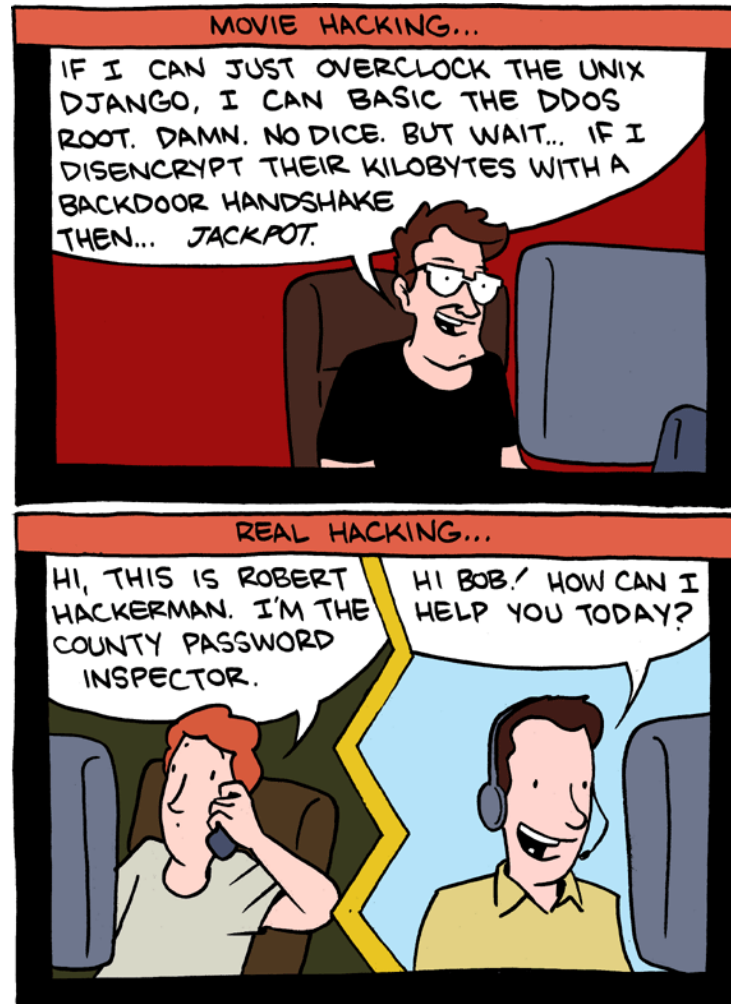


Agenda

- Introduction
- Target overview
- Attacks + demos
- Summary

Introduction

What's it about?



Source: <http://www.smbc-comics.com>

This is what it's about

- Practical attacks
- Likely to happen
- Easy to execute
- Great potential for paying off



Why USB modems?

- Very popular
 - ~130 million devices shipped in 2013
- Few vendors
 - Not that many models
 - Shared code between models



Target overview

Previous research

- Nikita Tarakanov & Oleg Kupreev
 - From China With Love (Black Hat EU 2013)
- Rahul Sasi
 - SMS to Meterpreter – Fuzzing USB Modems (Nullcon Goa 2013)

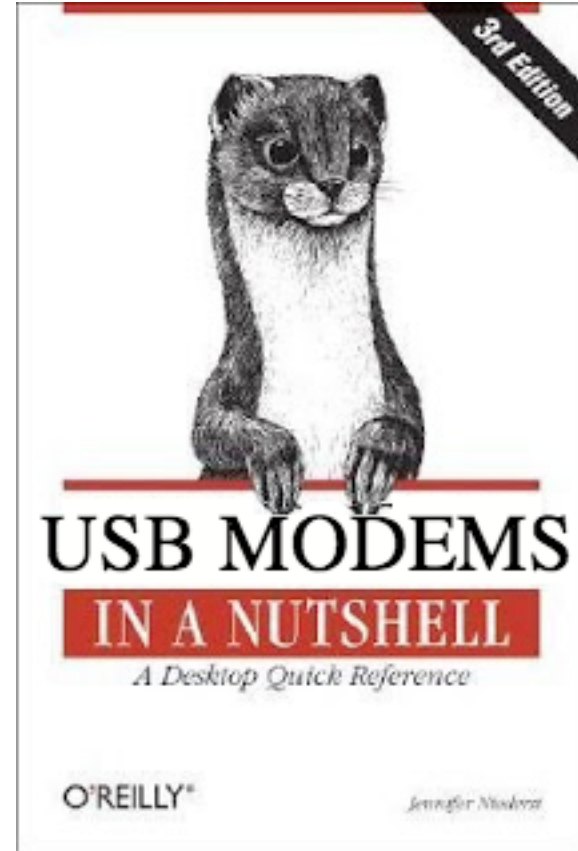
Scope

- Devices from the two biggest vendors*
 - Huawei
 - ZTE
- Focus on one device from each
 - Huawei E3276
 - ZTE MF821D
- Identify common attack surface

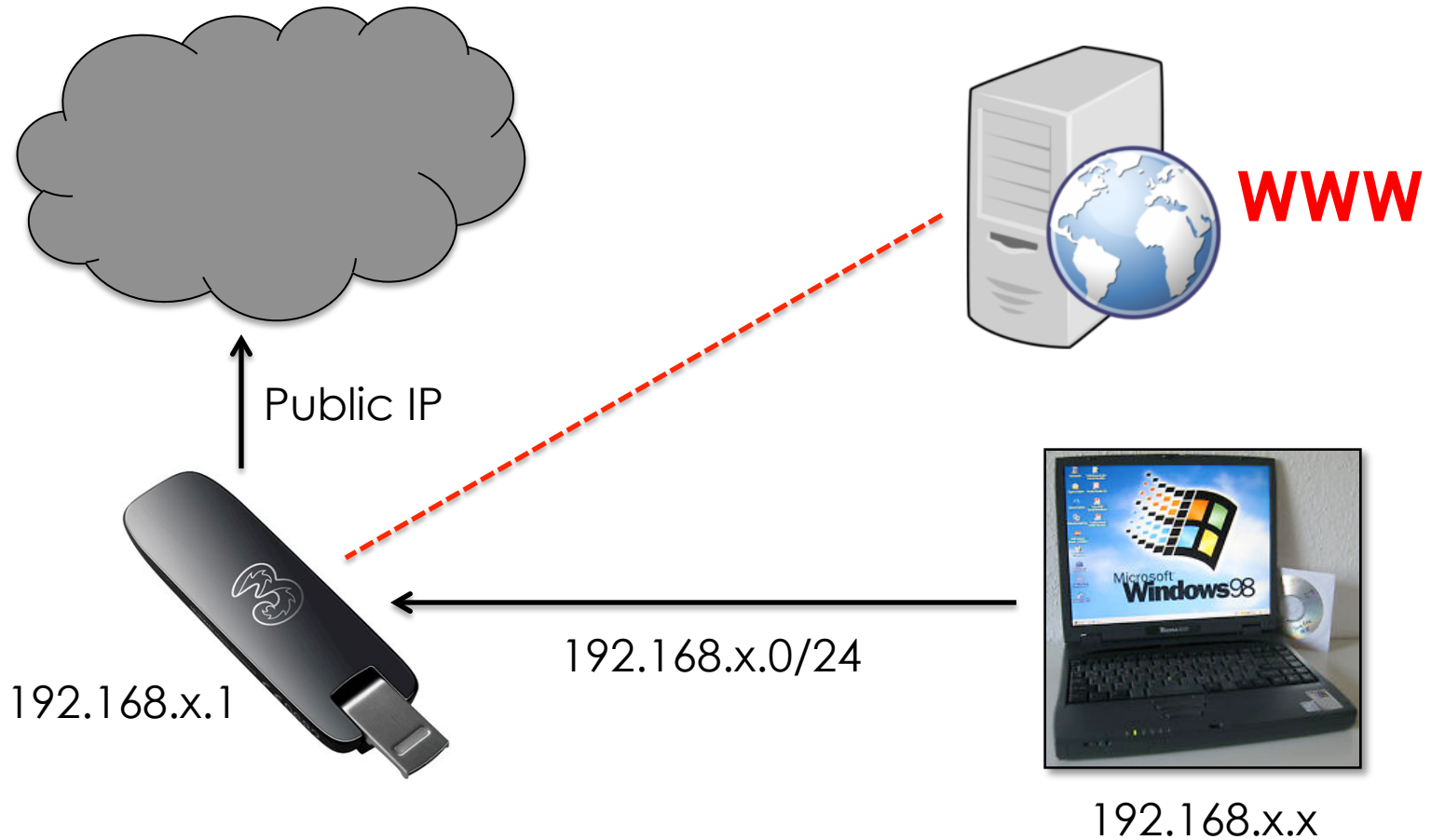
*Combined market share of more than 80% in 2011 (www.strategyanalytics.com)

In a nutshell

- Runs embedded Linux
- Mobile capabilities
 - GSM, 3G, 4G, SMS
- Web interface
 - Part of carrier branding
- No authentication
 - Single-user device



Network topology



Attacks

or

“What would Robert Hackerman do?”

Ground rules

- Objectives
 1. Make money
 2. Steal information
 3. Gain persistence
- Pre-requisites
 1. Remote attacks only
 2. See #1



Out of scope (but possible)

- Disconnect the device
- Lock out PIN and PUK
- Permanently break the application

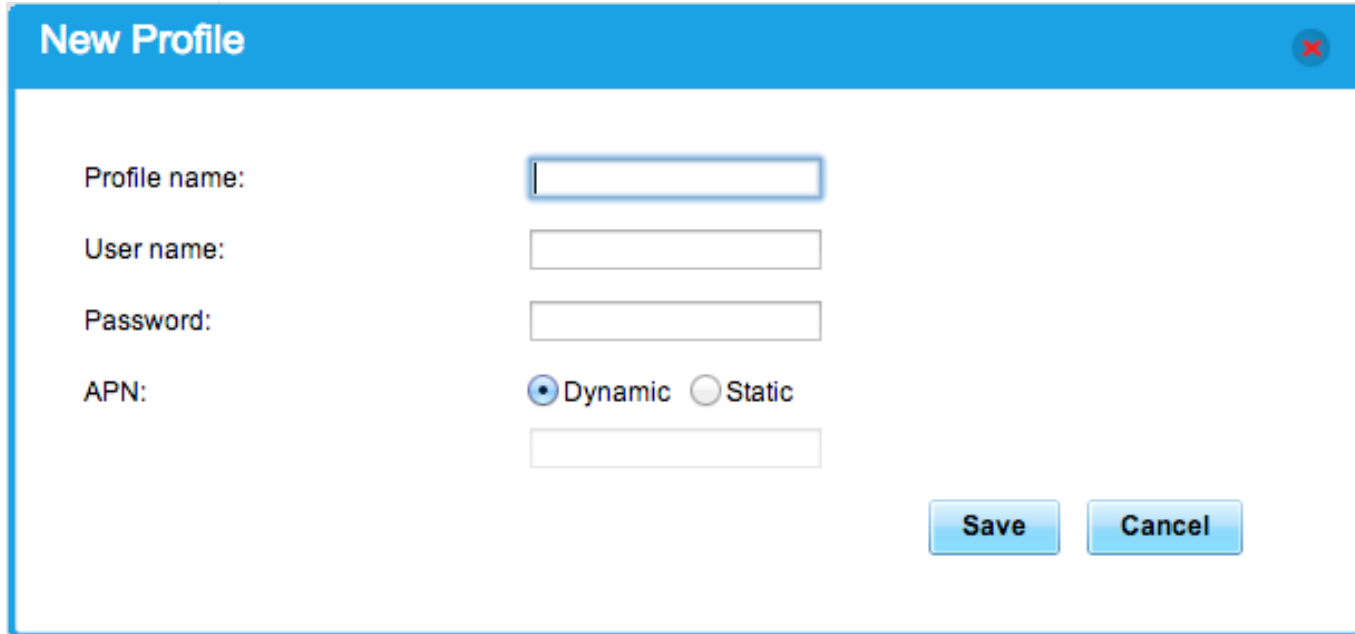
```
if((NetStat_roam!='2') && ((roamstatus_roam == '1')&&('2' == service_status_roam)) && (roam_status_info == '1')) {  
if(!confirm(On_roam[lang_index])) { flag  
service_status_roam)) && (roam_status  
top.global_set_cookie("disconnect_wai  
top.global_set_cookie("always_on_to_r  
file=4&node=AutoManual&value=1&rd=  
file=4&node=AutoManual&value=1&rd=  
document.netConnect.action = "/goform  
$(document).ready(function() { $("#manual_mode_show").click(function() { }); change_state(); setInterval("change_state()",1000);  
setTimeout("change_state()",1); }); function initpage() { get_pin_status(); initTranslation(); }  
am == '1')&&('2' ==  
_index]); } if(flag == true) {  
; if(Auto == '0') {  
de?  
/goform/SetNetworkSelectionMode?  
//alert("after"); }  
return false; } }
```



- Permanently brick the device

Attacking configuration

DNS poisoning



The image shows a 'New Profile' dialog box with a blue header and a white body. The dialog contains the following fields and controls:

- Profile name:** A text input field.
- User name:** A text input field.
- Password:** A text input field.
- APN:** A radio button labeled 'Dynamic' (which is selected) and a radio button labeled 'Static', followed by a text input field.
- Buttons:** 'Save' and 'Cancel' buttons located at the bottom right of the dialog.

DNS poisoning

```
<?xml version="1.0" encoding="UTF-8"?>
<request>
  <Delete>0</Delete>
  <SetDefault>0</SetDefault>
  <Modify>1</Modify>
  <Profile>
    <Index></Index>
    <IsValid>1</IsValid>
    <Name>malicious</Name>
    <ApnIsStatic>1</ApnIsStatic>
    <ApnName>internet.telenor.se</ApnName>
    <DialupNum>*99#</DialupNum>
    <Username></Username>
    <Password></Password>
    <AuthMode>0</AuthMode>
    <IpIsStatic></IpIsStatic>
    <IpAddress></IpAddress>
    <DnsIsStatic></DnsIsStatic>
    <PrimaryDns></PrimaryDns>
    <SecondaryDns></SecondaryDns>
    <ReadOnly>0</ReadOnly>
  </Profile>
</request>
```

DNS poisoning

- CSRF to add a new profile
- Static DNS servers
- Read Only & Set Default
- Remove original profile
- Send user to ad-networks, malware sites, spoofed websites, etc.

DNS poisoning - bonus attack

- Trigger firmware update
- Spoof update server
 - Downloads are over HTTP
 - No code signing
- Potentially get user to install backdoored firmware...



SMS MitM

SMS Settings

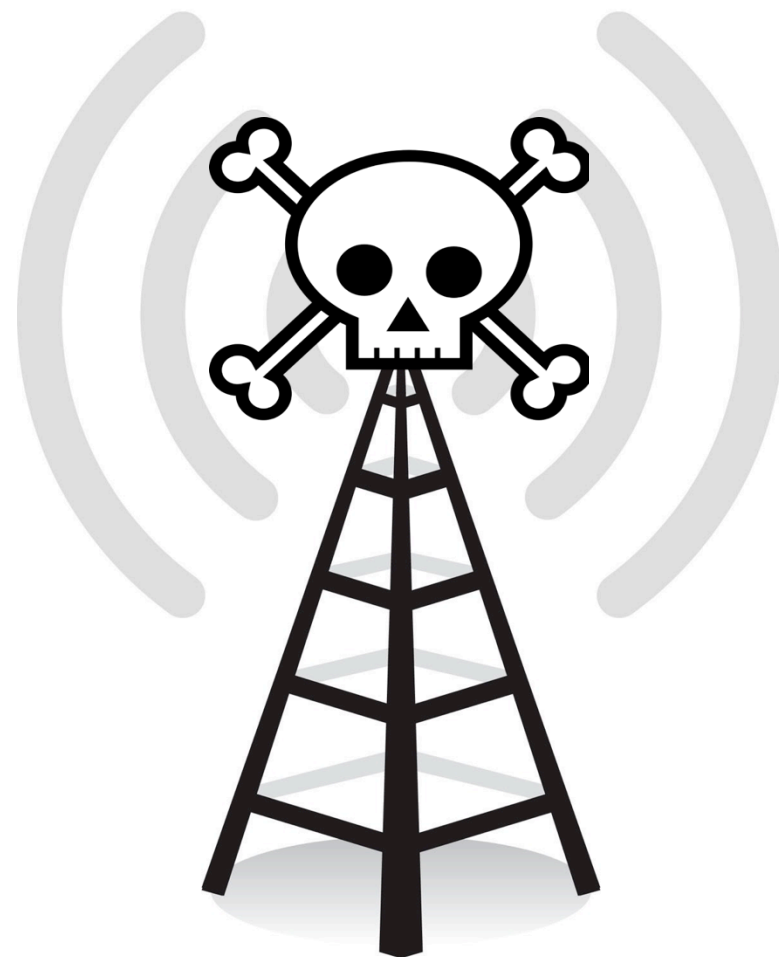
SMS report: Enabled Disabled

Apply

```
<?xml version="1.0" encoding="UTF-8"?>
<request>
  <SaveMode>0</SaveMode>
  <Validity>10752</Validity>
  <Sca>+46708000708</Sca>
  <UseSReport>0</UseSReport>
  <SendType>1</SendType>
  <Priority></Priority>
</request>
```

SMS MitM

- Replace the Service Center Address
- Set up rogue SMSC
- MitM all outgoing text messages



Abusing functionality

CSRF to SMS

- CSRF to make the modem send SMS
 - Send to premium rate number
- Potentially identify the user
 - Look up phone number
 - Twin cards
- Useful in targeted phishing attacks

Demo

Let's go phishing!

Getting persistent

Getting persistent

- Multiple XSS vulnerabilities
- Configuration parameters

```
data_roam_option=1';alert(1);//&submitRoam=Apply
```

```
function initTranslation2()  
{  
var roam_status_info = '1';alert(1);//';
```

- Configuration is persistent...

Getting persistent

- The web interface is where you go to connect to the Internet
 - Huawei Hilink opens main page automatically
 - ZTE creates a desktop shortcut
- The main page sets everything up
 - Loads an iframe for user interaction
 - It also loads the chosen language

Getting persistent

- Language is a configuration parameter loaded by the main page

```
function change_lang_cookie_before_mlang_js()  
{  
  var xml_lang = 'en'  
  setCookie('lang', xml_lang, 60*24*20);  
}
```

- It is injectable...

```
GET /goform/web_upd_xml?file=4&node=Language&string=en&rd=
```

Getting persistent

- Execute code every time the user connects to the Internet
- Interact with injected code
- Command channel
 - Poll remote server (BeEF style)
 - Out of band over SMS

Demo

SMS hooking

Summary

What to expect

- Attacks on configuration
 - Network
 - Mobile
- Abuse of functionality
 - Outbound & inbound SMS
- Injection attacks
 - Getting persistent
 - Stealing information

Getting it fixed

- ZTE is “working on it”
 - I have no details
 - ZTE does not seem to have a product security team ☹️
- Huawei is fixing their entire product line
 - Nice++
 - Huawei has a product security team 😊
- Sounds pretty good though, right?

The update model is broken


- Vendors cannot push fixes directly to end-users
 - Branding complicates things
- Vendor -> Carrier -> User
 - Carriers might not make the fix available
 - Users might not install the fix
- Most existing devices will probably ***never get patched***

Summary: analysis

- Web is easy
- Web is hard!
- How about the Internet of Things?




OWASP Internet of Things top 10



OWASP
Open Web Application
Security Project

The OWASP Internet of Things Top 10 2014 (tentative) is as follows:

- I1 Insecure Web Interface
- I2 Insufficient Authentication/Authorization
- I3 Insecure Network Services



Don't forget...



Marcus J. Carey @marcusjcarey · Jun 7

Risk associated with a lot of the "trending" security research is ridiculously small. Just ignore the stuff we can fix. [#nothingtoseehere](#)

Details

← Reply ↻ Retweet ★ Favorite ... More



Andreas Lindh @addelindh · Jun 7

[@marcusjcarey](#) 100% agree - stunt hacking sells tickets, real bad guys pick low-hanging fruit. pic.twitter.com/H8VMVNNCwG

Details

← Reply ↻ Retweet ★ Favorite ... More

Thank you for listening!

Andreas Lindh, @addelindh, Black Hat USA 2014