



Presented by:
Jeremy Allen
Rajendra Umadas

April 21, 2011

Network Stream Hacking with Mallory

- **Who are we?**
 - Rajendra Umadas
 - Jeremy Allen
- **What do we do?**
 - Mobile Application Assessments, Thick Clients, COTS (lots of other assessments)
- **What annoys us?**
 - Proxy setup
 - Throwing away Tools
 - Wasted or Redundant effort
- **What did we do about it?**
 - Mallory!

Who Are We

Jeremy Allen

- Principal Consultant
- iOS Assessment Lead
- Writes a lot of Python
- Try to break things before the bad guys do

Rajendra Umadas

- Consultant
- Mobile Application Hacker
- Does not afraid of anything
- Mobile Application Security = fun;

What we do

Application Assessments

- **Mobile Applications**
 - Acronym and Name Overload:
QUALCOMM/BREW, RIM, Windows Mobile,
Windows Phone, iPhone, Android ...
- **Web Applications**
 - XSS, SQL Injection – Pays the bills, still how many
get owned
- **Other Apps**
 - Thick clients, binary protocols, “harder” targets

HTTP Proxy

- When is a HTTP Proxy Not Enough?
- Protocols that just use HTTP for transport
- Unidentified network streams in tested application

Other Tools

- What tools exist that are good for MiTM and app testing for application assessments?
- dsniff, cain, tcpdump ,wireshark (about that...), others

Solution

“The enemy knows the system” –
Claude Shannon

Hard Way

- ARP
- DNS Spoofing
- iptables and custom scripts
- Packet level techniques

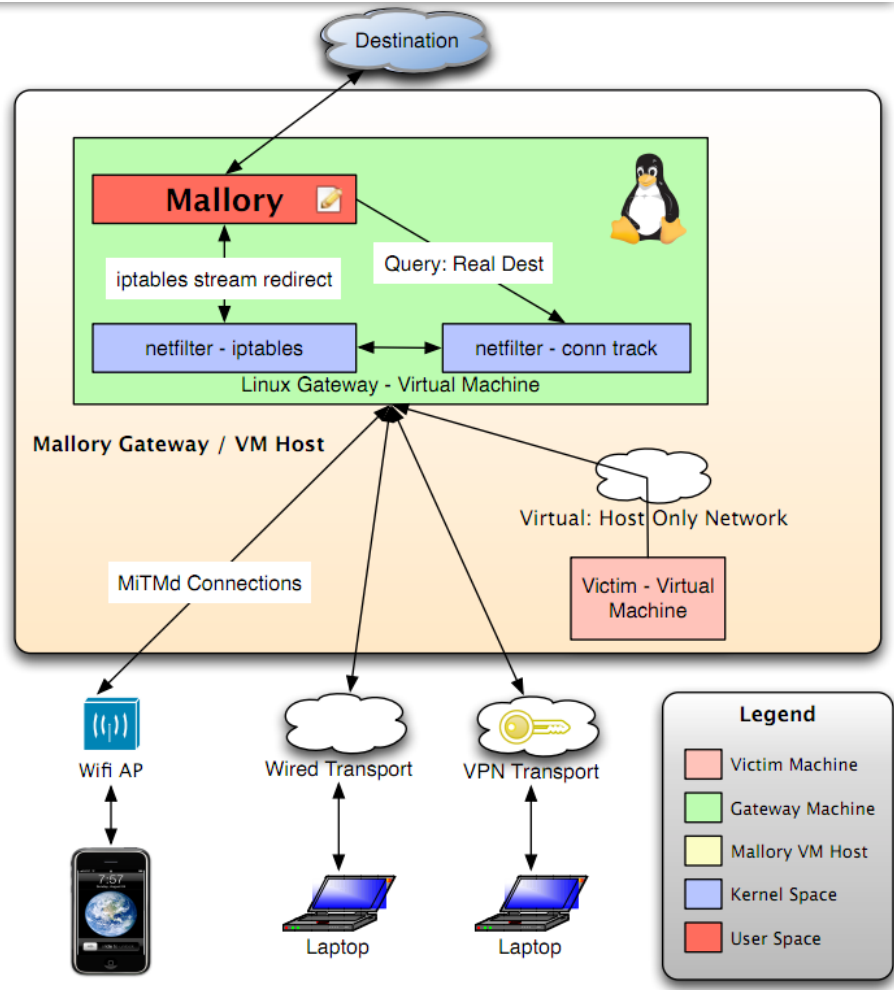
Easy Way



Demo

- Intro to Mallory GUI

Solution: Mallory Architecture



What have others done with Mallory?

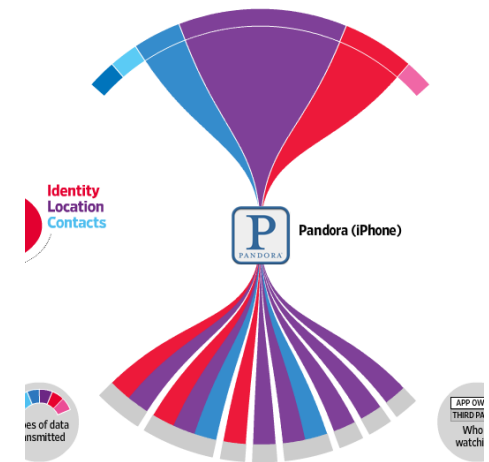
- **WSJ Application Privacy Research**

About data categories
Phones can collect a trove of data about the user. The Journal looked at the key categories below.

User name, password
Contacts
Age, gender
Location
Phone ID
Phone number

MOUSE OVER FOR MORE

THE WALL STREET JOURNAL.



Application Testing

- **WSJ Application Privacy Research**
- **Chose Mallory because it is pervasive**
- **Apps don't matter, Mallory can see the traffic**

Demos + Deep Dive



GUI Configuration

- Common tasks are fast, easy (configuring mallory, editing TCP streams)
- Uncommon tasks are possible, fairly easy (rules, known protocols)
- Rare tasks are harder, require some trickery (python plugins, python code)

GUI Configuration

- Getting traffic to the stream editor
- Echo Server
- Rules Config Demo

Demo – Something More Serious

- TCP Stream Editing
- All flows and datagrams go to a database
- Show database in advanced view

Programmatic Modification

- What about places where editing is harder?
- DEMO – Modifying VNC Keystrokes

What about Fuzzing?

- Fuzzing Paradigms
- File fuzzing
- Network fuzzing

How does Mallory Fuzz?

- ProxFuzz
- Opportunistic Fuzzing
- Protocol Depth

How does Mallory Fuzz?

- DEMO

Fuzzing Results

- Can vary greatly
- Depends on your app
- We have seen good returns with opportunistic fuzzing

Common App Testing Problem

- SSL

Mallory Knows SSL

- DEMO - SSL

SSL Usage

- Mallory acts as a CA
- Generates Certs in Memory
- Makes Look Alike Certs

SSL Usage

- Mallory acts as a CA
- Generates Certs in Memory
- Makes Look Alike Certs

Conclusion

Conclusion

- Mallory is a MiTM Proxy
- Focus on streams and datagrams
- Mallory can decode and MiTM: HTTP, HTTPS, SSL, DNS, SSH
- Mallory make common app testing tasks easy