

- What is John
- Create a benchmark and get a baseline
  - Set up an Ubuntu VM, create 5 users with various passwords
  - Test these passwords on Tufts' homework and pulsar servers overnight.
  - See what I get back in a 12 hour period
- Look at existing options
  - Djohnd
- Exploring the distributed options of JtR
  - Manually create distributed instances of JtR across 116 or 118
  - Change the incremental mode for each machine
  - See what I get back in a 12 hour period
- Scripting!
  - Write a script that will find available machines on a network
  - Count the number of available machines
  - Require a list of types distributions to use
    - Allows user to specify what they think will be the ranges of passwords
      - Lowercase alpha, 5 characters
      - Uppercase alpha, 5 characters
      - Mixed alpha, 5 characters
      - Mixed case alphanumeric, 0-5 characters
      - Etc, etc
  - Run individual john processes across discovered machines
- Brief look at alternatives to JtR
  - oclHashcat

## Sources:

<http://openwall.info/wiki/john/parallelization>

<http://ktulu.com.ar/blog/projects/djohn/>

<http://www.admin-magazine.com/Articles/John-the-Ripper>

<http://hashcat.net/oclhashcat-plus/>

Remember,  
this is  
a paper  
too!

Ayal Pierce

## Outline: Homomorphic Encryption in Voting

Abstract

Introduction

Why care about voting?

- Voting is the foundation of democracy
- Technology is evolving but voting is not.
- 2000 presidential election controversy: Highlighting how many of the issues present in 2000 still can potentially occur today.
- Help America Vote Act (2002): Replace punchcards and lever voting systems, started the move to electronic voting

The Ideal Voting System and problems with current system

- Accuracy: Winner needs to win
- Privacy: Votes should be private
- Individual and Universal verifiability without a readable/unencrypted receipt (as to avoid coercion).

Problem with the current system

- Paper ballots have hanging chads and are often confusing
- Electronic voting have no paper trail
- Electronic voting has no verifiability and is subject to bugs (government technology is not always the best and most secure)

How will the system work?

- Pret a Voter ballot (as opposed to the bingo ballot)
- Scratch off void box, for optional auditing
- Homomorphic Encryption
- Cost

Proposed Alternatives

- Mixes (rather than homomorphic encryption)
- Invisible Ink "Scantegrity II" verifiability

Potential attacks and flaws on the new voting system

Conclusion

Some References

- Adida, Rivest, et al (*Scratch and Vote, self-contained paper based cryptographic voting*).
- Xia, Schneider, et al (*Analysis, Improvement and Simplification of Pret a Voter with Paillier Encryption*)
- Ryan, Bismark, et al (*The Pret a Voter Verifiable Election System*)
- Mora, Naor, et al (*Split Ballot Voting: Everlasting Privacy with Distributed Trust*)
- Lovgren (*National Geographic: Are Electronic Voting Machines Reliable?*)
- Chaum, Carback, et al (*Scantegrity II: End-to-End verifiability by voters of optical scan elections through confirmation codes*).

Ideas for Supporting Material: Prezi Presentation, informational website

this should be 1/2 of paper

✓

what is this?  
focus on this

Nicholas Andre  
iClass RFID Access Control System Vulnerabilities  
Security Paper

Overview of ideal access control system => What an ideal iClass system would provide

- Physical token that cannot be duplicated
- Physical token is the only way to gain access to access points => no back doors
- Can be deactivated if lost => intelligent authorization of users
- Not possible to get a token from a third party that could be used in place of a token
- Can be given time based access

Comparison to traditional methods:

- Keys are fairly easily duplicated personally or from a third party. The only exception are "patented" blanks or some microchip keys, but most are still duplicatable using 3D printers or commercially available duplicators.
- Locks are somewhat easily bypassed by picking, bumping, or brute forcing
- It is very difficult to design a time based physical lock or to revoke access entirely for a single person to all locks in a given set.
- Tiered access systems are more difficult (master keyed systems) and are typically no more than two or three tiered. Invalidating a master key still requires changing every physical cylinder.

iClass features, according to marketing literature:

- ID cards use two way encryption to prevent sniffing the key off the air
- Two keys used to read the card: authentication and encryption key.
- Authentication key: used to release the data from the card -> without it, the card will not communicate.
- Encryption key: if it is incorrect, the card "ID number" will be scrambled
- Computer based backend will allow any number of scheduling
- Cards provided by iClass directly with customer keys are not editable by the customer without knowledge of the key in readers/writers
- No third party cards available.

iClass system problems:

- Included mechanical sensors/actuators can easily be set up to provide further back doors into the security of access points. These would not be as obvious as a broken lock in a traditional access system. For example, an Arduino with a power supply could easily apply a voltage to the unlock system for any momentary on lock system. This would be very difficult to detect and a near permanent defeat of the system.

iClass card problems:

- Card memory is writeable. You can change the ID number.
- Card hardware ID is not transmitted to the backend.

- Cards are readily available in sufficient quantities (albeit somewhat expensive).

#### iClass reader problems:

- Card readers are encased in epoxy, but the PIC microcontroller's ICSP connectors are exposed, presumably for testing or later upgrades by iClass.
- Even though "code protect" bits are set on the PIC microcontrollers to prevent loading the operating code out of the flash memory, it's possible to overwrite portions of the code in the bootloader etc to cause the reader to print its memory contents (requires two readers and a lot of time investment)
- The exposed ICSP port allows PIC DMA debugging, literally allowing the debugger to hand feed instructions to the instruction register. The external debugging system also controls the clock when this happens.
- Using debugging, one can easily freeze operation of the reader and dump the entire contents of RAM off the debugging port in a few seconds.
- Included tamper sensors aren't enabled by default.
- The RAM includes both keys AND the data of the last card to tap (regardless of whether the backend allowed access to the card's control system).
- The included "security screw" would prevent using normal screwdrivers to remove the reader; most of the time installers don't bother to add this extra layer of protection.

#### Attack outline:

- Locate reader out of plain sight.
- Use off-the-shelf phillips head screwdriver to remove the reader, while powered on.
- Defeat tamper sensor using electrical tape.
- Plug in pre-fabricated microchip or usb powered FTDI chip. Activate download and retrieve both keys necessary to duplicate
- Replace reader.
- Use a readily available iClass USB card writer to write the associated keys and ID number to another iClass card. This creates an EXACT COPY of the iClass RFID card--indistinguishable from the authorized user from the point of the system backend.
- You gain the access of the last person to use this reader. For example, if this reader were out of sight and typically only used by administrators who have a very high access clearance, you would gain the access level of that administrator until the system managed to detect the unauthorized activity and deactivate both the card.
- The only way to detect unauthorized access is some sort of time related violation--either the user taps on two readers which are physically far away during a short period of time or the user taps in at an unusual time. Automated methods intended to detect this sort of unauthorized usage may lead to authorized users losing access if they visit a different part of the system or enter at unusual times.

#### Applicability to Tufts:

- Tufts uses cards provided by iClass preloaded with the encryption keys. The readers have those keys loaded into hardware. The readers allow Tufts public safety officials to

read keys and load the ID into the authorization database but not write keys, which may lead to the impression by the system administrators that the cards are NOT easily duplicatable (while in fact they are).

- There are a number of “off the beaten path” readers frequently used by only campus police officers with campus-wide access.
- Has a 6-7 figure investment in a fatally flawed security infrastructure protecting nearly everything on campus.
- Administrators are most likely unaware that this vulnerability exists.
- This represents the flaw with relying on closed source and proprietary security infrastructure. Without extensive independent third party reverse engineering and penetration testing, it's very difficult to say for certain whether or not a system like this is secure. Administrators can only hope for the best.

How do you close  
the security  
holes?

all nearby vessels (essentially a denial of service attack). This can also be tagged to a geographical area e.g. as soon as ship enters Somalia sea space it vanishes of AIS, but the pirates who carried out the attack can still see it.

- Fake a “man-in-the-water” distress beacon at any location that will also trigger alarms on all vessel within approximately 50 km.
- Fake a CPA alert (Closest Point of Approach) and trigger a collision warning alert. In some cases this can even cause software on the vessel to recalculate a course to avoid collision, allowing an attacker to physically nudge a boat in a certain direction.
- Send false weather information to a vessel, e.g. approaching storms to route around.
- Cause all ships to send AIS traffic much more frequently than normal, resulting in a flooding attack on all vessels and marine authorities in range.

All of this is made possible because the AIS protocol was designed with seemingly zero security considerations. In particular, we noted the following major issues:

- Lack of Validity Checks: It is possible to send an AIS message from any location for a vessel at another location e.g. you can send a message from a location near New York for a vessel that claims to be in the Gulf of Mexico, and it will be accepted without question. No geographical validity checks are carried out.
- Lack of Timing Checks: It is also possible to replay existing (valid) AIS information, because no timestamp information is included in the message e.g. you can replicate the position of a vessel.
- Lack of Authentication: There is no authentication built into the AIS protocol. That means that anyone who can craft a AIS packet can impersonate any other vessel on the planet, and all receiving vessels will treat the message as fact.
- Lack of Integrity Checks: All AIS messages are sent in an unencrypted and unsigned form, making them trivial to intercept and modify.

## Conclusion

- During the peak of piracy in the Atlantic Ocean, pirates would frequently masquerade under false colors until the last moment. Their ships were stolen, so visual contact did not help much.
- This is a different age.
  - Backup systems exist
    - Radar
    - Satellite
    - Radio
    - Visual
  - For piracy, the boats used are very identifiable, and the security risk of piracy is much more a factor of avoiding the area.
- As for the few cases where these vulnerabilities could pose serious threats, we do need to look to investing in modern infrastructure. AIS has remained virtually unchanged since its adoption. Just because ingenious ways have not been adopted does not make the system good. It is insecure as can be, and in combination with other tools, the exploitation of AIS may be a larger problem if we do not consider a newer standard.

How do you close security holes?

## Outline for Comp 116 Final Paper

Advisor: Ming  
Hao Wan

### 1. Abstract

- a. summarize the main ideas of this paper: security of using hotspot WIFI connection via mobile devices.

### 2. Introduction

- a. Briefly introduce the concept of WIFI/hotspot.
- b. Explain the improved security of HTTPS over HTTP with a focus on SSL(Secure Socket Layer).
- c. Expand on the vulnerabilities of public WIFI connection even with SSL and explain a few interesting attacks: CRIME and BREACH attack, Man-in-the-Middle attack, Renegotiation Attack, etc.

### 3. To the Community:

- a. Comment on the increasing popularity of public WIFI, on the background of the increasing number and usage of mobile devices.
- b. Emphasize on the potential problems caused by insecure public WIFI access: Man-in-the-Middle-Attack, Session Hijacking, Sniffing, etc.
- c. Comment on how secure practices can have potential benefits to the community.

### 4. Action Items/Defenses/Applications

- a. Explore the way to conduct attacks such as CRIME and BREACH attack on mobile devices (phone devices preferred) connected to a public WIFI.
- b. Explore methods to protect mobile devices from attacks through public WIFI.
- c. Potentially conceptualize a way to improve the WIFI connection security

### 5. Conclusion

- a. Reiterate the abovementioned points with a focus on section 4.

### 6. References

- a. Security Mechanisms and Security Analysis: Hotspot WLANs and Inter-Operator Roaming  
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1391371&tag=1>
- b. Enhancing Web Browsing Security on Public Terminals Using Mobile Composition <http://dl.acm.org/citation.cfm?id=1378612>
- c. Method and system for privacy in public networks  
[http://www.google.com/patents?hl=en&lr=&vid=USPAT7185204&id=QxB\\_AA AEBAJ&oi=fnd&dq=public+wifi+security&printsec=abstract#v=onepage&q=](http://www.google.com/patents?hl=en&lr=&vid=USPAT7185204&id=QxB_AA AEBAJ&oi=fnd&dq=public+wifi+security&printsec=abstract#v=onepage&q=)

public%20wifi%20security&f=false

- d. Characterizing Privacy Leakage of Public WiFi Networks for Users on Travel [http://spirit.cs.ucdavis.edu/pubs/conf/Ningning\\_INFOCOM13.pdf](http://spirit.cs.ucdavis.edu/pubs/conf/Ningning_INFOCOM13.pdf)
- e. Security for Mobile ATE Applications  
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6334538>

7. Idea Supplemental Material:

- a. Video demonstration featuring public WiFi exploitation in a Cafe. The video will refrain from infringing on other people's privacy by focusing the attack on the researcher's own device.

Perhaps  
mention  
how many  
mobile  
devices  
autoconnect  
to wifi



Is this possible?

True Random Number Generation for Cryptography, on the Cheap

Final Project Outline for Comp 116

Paul Nixon

pauljnixon@gmail.com

Mentor: Ming Chow

Why it's important

- The revelation that the NSA has been asking corporations for cryptographic keys suggests that they can't crack RSA algorithmically.
  - So, it's important to prevent against non-algorithmic attacks on cryptography.
  - Maybe these 2 points aren't actually connected
- Explanation of difference between "True" Random Number Generators and Pseudo Random Number Generators.
  - Essentially, the use of some entropy source outside of the computer
- Examples of bad cryptography via bad randomness
  - Debian SSL debacle
    - Use of uninitialized data produced warnings, but it was an important randomness source
  - Due to the PRNG not being seeded with uninitialized data, there were only 32768 possible keys (or less, the only other entropy was Process ID)
  - NSA backdoor in the Dual\_EC\_DRBG PRNG

Other Solutions

- Other True Random Number Generators exist
  - Clock Drift
    - Truerand
      - Most famous implementation of clock drift, but flawed by its own admission
  - Analog noise
    - Random.org provides random numbers from atmospheric radio noise for free to anyone
      - Actually based on the same principle as mine, just more advanced
    - SGI pointed webcams at lava lamps

# Comp 116 Final Paper Outline

Stefan Dimitrov

October 31, 2013

## Introduction

Problem: iOS jailbreaking has been stigmatized as a potential invitation for malware and security risks, while too much trust is placed in the walled garden model of software distribution through the App Store. How can we turn jailbreaking into an advantage?

## To the Community

1. Jailbreaking allows for:
  - unrestricted security research on the iOS platform.
  - third party patches in response to security vulnerabilities.
2. This paper:
  - points out some security and privacy benefits of jailbreaking
  - uncovers pitfalls that could make jailbreaking a security risk

## Privacy issues

1. UDID leaks [b]
2. Path address book fiasco
3. Apps sharing too much for no reason other than identification, data collection, etc.
4. ...

## Security issues of stock-iOS devices

1. Methods for subverting the signing process on non jailbroken devices
  - a GBA emulator's source code used to be available on github and could be compiled and installed without a developer account. ad hoc distribution using leaked uids

2. Methods for injecting obfuscated malicious code into an App Store app
3. Other methods of delivering malware
4. ...

#### Security issues of jailbroken devices

1. Malicious/untrusted repositories
2. SSH attacks with default root password
3. Retaining a jailbreak means not updating to the latest iOS
4. ...

#### Security benefits of jailbreaking

1. Delivery of patches (jailbreak.me fix for pdf vulnerability)
2. Allows for security research, finding flaws and patching them
3. Enhancing privacy - on stock iOS7 camera access still unrestricted, ad location tracking [a]
4. Prevents *other* users from jailbreaking for malicious purposes (bypassing the lockscreen passcode lock for example) [c]
5. ...

#### Summary

#### References

##### iSAM: An iPhone Stealth Airborne Malware

[http://www.icsd.aegean.gr/publication\\_files/conference/62773319.pdf](http://www.icsd.aegean.gr/publication_files/conference/62773319.pdf)

##### PiOS: Detecting Privacy Leaks in iOS Applications

<http://seclab.cs.ucsb.edu/media/uploads/papers/egele-ndss11.pdf>

##### Jekyll on iOS: When Benign Apps Become Evil

<http://www.cc.gatech.edu/~klu38/publications/security13.pdf>

##### A survey of mobile malware in the wild

<http://dl.acm.org/citation.cfm?id=2046618>

##### Exploiting the iOS Kernel

[http://media.blackhat.com/bh-us-11/Essex/BH\\_US\\_11\\_Essex\\_Exploiting\\_The\\_iOS\\_Kernel\\_WP.pdf](http://media.blackhat.com/bh-us-11/Essex/BH_US_11_Essex_Exploiting_The_iOS_Kernel_WP.pdf)

## Articles

[a] <http://ios.wonderhowto.com/how-to/18-sneaky-privacy-betraying-settings-every-iphone-owner-must-know-about-ios-7-0148682/>

[b] <http://www.crowdstrike.com/blog/finspy-mobile-ios-and-apple-udid-leak/index.html>

[c] <http://www.zdnet.com/ios-7-apples-war-against-jailbreaking-now-makes-perfect-sense-7000016623/>

## Code Project Idea:

Write a daemon that monitors for unauthorized filesystem access, data leaks, etc.

So this is like  
On the phone?

## Introduction

Thesis (I don't know what this is yet!)

What is metadata?

yes

## Availability

Metadata that is freely available

Metadata that is available to someone with technical knowledge and few scruples, i.e. attackers willing to break the law.

Metadata that is available to government agencies

## Analysis

What sorts of knowledge can be achieved by manual examination of metadata?

What can be learned by looking at large to enormous sets of metadata, through statistical analysis and other "traditional" techniques?

A discussion of my own supporting materials, which will be a program in the latter category. Hopefully this will yield interesting results which will provide for ample discussion.

## Conclusion

No idea, I haven't come to any conclusions yet.

## Supporting materials

I am particularly interested in how much data can be inferred about an individual from the information they leave lying around. Many of us take at least some care not to leave such a fingerprint on the web that say, comments in forums could by a clever investigator be linked to an individual. But the world is small, and there's a lot of data. Metadata is particularly interesting because we're much less aware of it. Obviously if we leave our email address in a forum comment we're aware that some anonymity is lost. But what about the geotags on the pictures we post, the times of day we're most active, the forums we post in. With enough data and a good algorithm, there is clearly a lot that could be inferred about the person behind the username.

Reddit is ripe for this project, because it has an enormous userbase, many of whom (I hope) may be somewhat cooperative. My idea, as it currently stands, is to begin by asking people to contribute real data; nothing incredibly personal, but things like hometown, first language, age, marital status, employment status; as much or as little as each is comfortable with associating with their reddit username. I would randomly divide this into two groups: a training group and a test group. My program would be a machine learning algorithm (I'd really like to try using a neural network, as they've always fascinated me, but I don't know a lot about them or

Need to  
be  
more  
thought  
out

What are they?

# Final Paper Outline

## Network Security

Jon Dotko

### A Technical History of Trans-Pacific Cyberwarfare

#### Introduction:

- The brief history of Sino-American geopolitics
- Imperialism & Boxer Rebellion
- World Wars
- Globalisation & Contemporary Espionage

#### The American Presence

- The NSA, and America's Security Protocol
- USAF Pacific cybersecurity protocol
- Industrial Espionage and network penetration

Very interesting

#### The Chinese Presence

- Chinese responses and adaptations
- Late adoption, swift entrenchment
- Across-the-pond retaliation

#### Entanglement

- 1995 → Present
  - \* Attacks & Engagement
  - \* Shifting security protocols
  - \* Political repercussions

#### Deliverables:

American and Chinese methodologies of attack  
(code samples)

#### Conclusions

- Forecasts
- Personal privacy implications
- How to influence American sec. policy now.

George Aquila  
Comp 116 - Computer Security  
10/31/2013

Topic – Cyberwarfare and the Stuxnet bug:

**Abstract:**

My final paper's topic will focus on the rising prominence of cyberwarfare and cybersecurity in international relations, specifically with regards to how it is used by countries like the United States to undermine its potential enemies. The paper will explore the nature of cyberwarfare on an international level, its potential as an actual weapon as war, and to what extent it will affect the growth of general computer systems security in other contexts. The specific example of how cyberwarfare has been used will focus on that of the the Stuxnet worm. The paper will explore the technical specifics of what made it such a unique, effective and devastating attack, how it was collaborated on in different contexts, and what the fallout of its leakage onto the world wide web implied.

**Background on the Stuxnet Bug**

It is no secret that for a number of years the United States and its allies have been attempting, by any means necessary, to stop the nation of Iran from acquiring nuclear capabilities with which it could potential manufacture weapons of mass destruction. Some time in mid 2009 or earlier, the United States added cyberwarfare to the list of these means in an effort to shut down Iran's nuclear program. The cyber attack came in the form of the Stuxnet worm, a malware program that was designed to attack a type of industrial systems (specifically those developed by Siemens, Simatic Factory System). This worm made it's way into the network of Iranian nuclear testing laboratories and reactors and targetted specifically the centrifuges necessary for carrying out nuclear fission.

-Effectiveness of the worm, both in terms of its devastating and rapid spreading, as well as its burrowing nature that makes it so hard to detect, has many implications for international cyberwarfare. The core idea here is that it may be more effective and useful to carry out attacks with burrowed worms of this nature rather than attempt a classic remote active assault on a system that is being actively monitored, as this attack is far more easy to detect than something like the Stuxnet worm.

**The Technical Side/What it Does**

-The Stuxnet worm is a highly specialized, highly targeted malware program which attacks a specific type of industrial system. This worm is first transmitted through a portable USB, being approximately half a megabyte, and can move quickly through any windows system due to its promiscuous behavior.

-The worm installs itself into the root of Windows machines, and disguises itself as a core driver, which is what allowed for it to avoid detection for the amount of time that it did. What makes it particularly powerful and devastating is its ability to spread any copy itself over a network very quickly, jumping from machine to machine with a simple

-Stuxnet takes advantage of several Windows exploits which allow it to copy itself, gain root

privileges, and bypass security barriers such as firewalls (which it does using an exploit of internet explorer).

-The worm communicates its progress and information back back to a command origin by transmitting an encrypted http request to a seemingly innocuous url. It can also receive further instructions from its command and control, which can transmit requests to call functions already present in the worm, or send and load new instruction sets with updated additional functions.

Can you get snippets of stuxnet code?

### Supporting Material

The supporting material will be a proof-of-concept to demonstrate the way that the stuxnet worm will find its way into a system and attack it without being detected. This will include my writing of a simple program emulating the burrowing effects of the worm. This program will most likely be written in C, as the worm itself was written in C and C++, despite this being fairly uncommon for a malware bug of its nature.

### References and resources

-[https://www.symantec.com/security\\_response/writeup.jsp?docid=2010-071400-3123-99](https://www.symantec.com/security_response/writeup.jsp?docid=2010-071400-3123-99)

A report by Symantec elaborating on the worm's functionality, infection process, prevalence, distribution, and a protection that they developed against the worm

-*Stuxnet Dossier*, Symantec Corporation. Cupertino, CA, 2011. PDF.

A 69 page dossier on the known workings and behaviors of the Stuxnet worm published by Symantec

-<http://www.f-secure.com/weblog/archives/00002040.html>

A useful Q&A by F-Secure on the nature and of the stuxnet attack on Siemens Simatic industrial systems

-<http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&objid=43876783&nodeid0=10805583&caller=view&lang=en&siteid=cseus&aktprim=0&objaction=csopen&extranet=standard&viewreg=WWW>

A potentially helpful guide published by Siemens of how to identify and purge a stuxnet infection on an industrial system

-[http://www.kaspersky.com/about/news/virus/2012/Resource\\_207\\_Kaspersky\\_Lab\\_Research\\_Proves\\_that\\_Stuxnet\\_and\\_Flame\\_Developers\\_are\\_Connected](http://www.kaspersky.com/about/news/virus/2012/Resource_207_Kaspersky_Lab_Research_Proves_that_Stuxnet_and_Flame_Developers_are_Connected)

A published analysis by Kaspersky Labs that concludes that stuxnet and Flame, a malware program similar to Stuxnet which I will be discussing, seem to have been developed by the same team of engineers

-<http://www.zerohedge.com/article/deadf007-stuxnet-secret-weapon-attack-irans-nukes-virus-about-revolutionize-modern-warfare>

A half-technical half-policy focused report on how the Stuxnet worm has changed the nature of cyberwarfare



## High-Level Outline

### *Information Flow Control and Privacy*

#### **Abstract**

- Provide a brief overview of the topic
  - What is Information Flow Control?
  - Why privacy is important in the growing technology

As the Internet grows and technology advances, more people use their computers to fulfill everyday tasks such as banking, contacting others and shopping. This is a great step forward because of all the amazing things that can be done with the click of a button. However, many of these activities require private information in order to work. More data is being stored online than ever before and it needs to be protected. Common ways of securing data include firewalls, cryptography and using public/private keys, but nothing prevents information from propagating on. For example, Person A wants to send Person B a secret message that only B can read. However, A doesn't entirely trust B and wants to make sure that B cannot send this data out to anyone else. To do this, A can use information flow control where they can dictate the flow of data. This process uses a type system and labels to secure data from being accessed by malicious parties. With more personal data being stored on computers, people need an easy and effective way of securing their privacy. Information flow control (IFC) is a possible solution to this because of the fact that it can track outputs through a program and prevent leaks. In this day in age where privacy is becoming more important, protecting data is a top priority.

#### **Introduction**

- A more detailed view of IFC
  - How does it work (Basics of labels)
  - What languages implement it
- Privacy and how IFC relates

#### **To the Community**

- Chose this topic because it is not very well known but can provide a good solution to our growing concerns about privacy
- Many people are afraid of functional programming, but this is a great application of it
  - Gives a reason to possibly try functional programming
- Teaches the idea of labels on data and how maybe that can be used

#### **Applications**

- If this were to be implemented, could protect user data
- Big problem in Androids is the fact that many applications hold your data
  - Could think about making that data anonymous using differential privacy
- If IFC were to be used in emails, it would also be very helpful to privacy
  - Then other people could not snoop around (hello NSA) your stuff
- Might even be helpful as the world moves toward cloud computing!

#### **Conclusion**

*Related Works?*

I. Outline: Mobile Malware (still working on title)

- a. Intro
  - i. Misconceptions
  - ii. Future of mobile
  - iii. defenses
- b. Myths
  - i. Many people believe that mobile malware is a common problem but it's not.
    - 1. 1 in 100,000 apps downloaded has potential to cause serious damage [1]
    - 2. Most apps are downloaded from either the Apple App Store or Google's Play Store.
      - a. Apple is incredibly rigid in their review process
      - b. Play Store isn't curated as well as App Store but better than other third party sites
  - ii. Concessions
    - 1. Still a real threat
      - a. In it's worst form, mobile malware can [1]
        - i. Transmit data
        - ii. Take pictures and video
        - iii. Track location
      - b. This data is used for extortion, identify theft, ect
- c. The Future of Malware
  - i. Experts suggest that mobile malware is still in it's infancy
    - 1. Hasn't been around long
    - 2. Mobile platforms have only gained traction in the past 5-10 years
  - ii. Mobile Financials are increasingly popular
    - 1. Google Wallet
    - 2. Passbook
    - 3. Criminals will begin to focus on phones
  - iii. The Corporate Effect [2]
    - 1. Vulnerabilities in frameworks will lead to breaches in company data
    - 2. Info can be leaked out of employees phones
    - 3. Some say solutions can included heavily secured networks
    - 4. Malware is evading static analysis tools
      - a. Malicious payload wasn't executed until a certain level in a mobile game app [2]
    - 5. Static analysis tools just aren't that good [3]
      - a. Great for getting through tons of commercial code but can be avoided
- d. Conclusion

Get a  
real  
Android  
malware,  
give  
an  
analysis  
of  
it

- [1] <http://www.economist.com/blogs/babbage/2013/10/difference-engine-0?fsrc=scn/tw/te/bl/thethreatinthepocket>
- [2] <http://www.darkreading.com/mobile/catching-mobile-malware-in-the-corporate/240163044>
- [3] [http://www.darkreading.com/attacks-breaches/distributing-malware-through-future-app/240162315?itc=edit\\_in\\_body\\_cross](http://www.darkreading.com/attacks-breaches/distributing-malware-through-future-app/240162315?itc=edit_in_body_cross)

## GPS Spoofing

### Outline:

1. Introduction
  - 1.1 The Global Positioning System
  - 1.2 Structure of GPS
    - 1.2.1 Space Segment
    - 1.2.2 Control Segment
    - 1.2.3 User Segment
  - 1.3 Application and Utility to daily life.
2. GPS Systems
  - 2.1 Application of GPS in Industries
  - 2.2 Mechanism & Vulnerabilities
    - 2.2.1 Smart Phone GPS system
    - 2.2.2 Aircraft/Ferry GPS navigation system
3. Different kinds of spoofing
  - 3.1 Mobile Self GPS spoofing(faking ur own GPS)
    - 3.1.1 Apps that does the spoof
    - 3.1.2 Apps that can be fooled
    - 3.1.3 How to GPS spoof smartphone
  - 3.2 Navigation GPS spoofing(fake broadcasting signals)
    - 3.2.1 Case study: hacking a ship's navigation system
    - 3.2.2 How it is done
    - 3.2.3 Potential threats and utility
4. Countermeasures and Prevention
5. Supporting material: Mobile GPS spoofer  
Demonstration of existing GPS spoofer  
Or  
Some code that does the GPS spoofing on smartphone
6. Conclusion
7. References
  - 7.1 Radionavigation Lab at U of Texas at Austin  
<http://radionavlab.ae.utexas.edu>
  - 7.2 GPS References - US Coast Guard Navigation Center  
<http://www.navcen.uscg.gov/?pageName=gpsReferenceInfo>
  - 7.3 GPS.gov  
<http://www.gps.gov/>

← this was  
big  
news

# AN INVESTIGATIVE STUDY INTO THE PRIVACY OF IOS APPLICATIONS

MICHAEL SILVERBLATT  
MENTOR: MING CHOW  
TUFTS UNIVERSITY

## 1. ABSTRACT

This paper will look to investigate the privacy and security levels of using iOS mobile devices. Recent discoveries have shown that many popular web and mobile application developers profit off of selling their users' data to various people. These include marketing agencies who in turn sell that same data to other companies, advertisers who want to target people directly, and the government, which believes it has a right to collect as much information about its citizens as it wishes. In many cases, users are not even aware that these applications have access to this data. Using mitmproxy and mitmdump to intercept WiFi SSL traffic to and from an iPhone, I will track the traffic for potential privacy violations. I will then choose a few applications in particular to focus on and make a profile of their privacy violations, if any. I will also address the importance of having privacy online, and the reasons why certain applications want your personal data. As a supplement to this paper, I will begin development of a wiki-like online directory of privacy concerns for iOS and other applications. My hope is to shed light on an often overlooked yet vitally important

concern about mobile computing, and to encourage use of applications which are honest with you about use of your data.

## 2. THE IMPORTANCE OF PRIVACY

**2.1. NSA/Government.** Everything is being stored, only need a warrant to actually look at information, massive datacenters could get cracked

**2.2. Marketing Agencies.** Data about you is being sold for advertising, credit/background checks, and other purposes. This information can often be wrong, and there is no way for you to change it or even see what it says.

**2.3. Application Usage Logs.** Some applications use companies like Flurry to log usage patterns. This is done by immediately sending a post request to a server to indicate that an app has been opened or closed. While one can see the obvious benefit to a developer of having a record of every user's usage of their product, one presumably can also see the concern that comes with having every action one takes on an iOS device being recorded.

## 3. HOW APPLICATIONS ACCESS PERSONAL DATA

**3.1. Privacy Policies.** Unlike Android, which prompts a user to approve data access permissions before downloading an application, the privacy policies for iOS apps can be slightly harder to find. In fact, most people never see them before downloading and installing many applications.

**3.2. Availability to Developers.** This section will investigate the availability of personal data to application developers. I'd like to discover how much information an application can pull without notifying the user that anything is going on.

[3] <http://www.technologyreview.com/news/516416/study-shows-many-iphone-apps-defy-apples-privacy-advice/>

← prevention?

**3.3. Location Data.** Of perhaps the biggest concern for many people is the collection and storage of location data. According to many in the industry, the GPS on iOS devices can be accessed and logged regardless of whether the device is even powered on. This is of obvious concern for anyone who wants to avoid a government database of every single person's every move.

#### 4. ACTUAL APPLICATION DATA

Here, I will look into actual packets sent and received by my iOS device in order to discover which applications are maliciously sending personal data to servers for purposes other than the application's stated functionality. I will look to focus on applications with a wide range of use at this point, such as Snapchat, Instagram, Chase Bank and other highly popular apps.

#### 5. SUPPLEMENTAL MATERIAL

As a supplement, I want to create a web application to store my finding about specific iOS applications. I will then look to make this available on the internet for the community to contribute to.

#### REFERENCES

- [1] <http://www.wired.com/gadgetlab/2012/02/path-social-media-app-uploads-ios-address-books-to-its-servers/>
- [2] [http://mesl.ucsd.edu/yuvraj/docs/Agarwal\\_MobiSys2013\\_ProtectMyPrivacy.pdf](http://mesl.ucsd.edu/yuvraj/docs/Agarwal_MobiSys2013_ProtectMyPrivacy.pdf)

Look into  
Veracoders  
Adi iOS

**Outline:**

1. Introduction
  - a. Idea that all wires and all electronics give off an EM signal when used
  - b. This concept was investigated by NSA in 1960's - project TEMPEST
  - c. Van Eck Phreaking
    - i. Used to capture signals from CRT monitors
    - ii. In 2004, image from conventional LCD monitor recovered
    - iii. In 2009 French group recovered signals from wired keyboards
  - d. The capability to tap into these frequencies previously limited by hardware costs
    - i. \$10 RTL-SDR from china tunes from ~50MHz to ~2,200 MHz
2. To the Community
  - a. Includes a very wide range of frequencies
    - i. Police scanners, regular radio, keyboards, wireless microphones, mobile phones (GSM, 3G, LTE), satellite communications, location systems for avionics and marine, etc
    - ii. ANYBODY CAN LISTEN TO THESE NOW!!!!!!!!!!
    - iii. Van Eck Phreaking is at the hands of anyone!
  - b. Melissa Elliot from Veracode did a DEFCON presentation on this subject
    - i. Shows how everything sends a signal – even the real time clock...
    - ii. Tries to recover signal from LCD monitor
    - iii. Shows that iPhone connects to 3G even in airplane mode!!
  - c. About the technology
    - i. More and more signals are digital, and some are encryptedⓈ
    - ii. Hard to decode the digital data to completely recover the signal
      1. Don't really need to decode the signal to get an idea of what is going on
        - a. Idea of correlated emissions
          - i. When the EM emissions from your device are correlated to an activity.
            1. Don't need to actually decode the signal to determine what state your device is in
        - b. i.e. when snooping on screen signals
          - i. The more complex the image on the screen, the more complex the signal
            1. Can be used to determine if the computer user is actively using his/her computer
    - c. RTL-SDR can be used to track people
      - i. Mentioned by Melissa Elliot

- ii. Can use emissions to determine what devices people are carrying
- iii. If

**Supporting Material Idea:**

1. To design a program that is able to recover the keyboard strokes from a wired keyboard using the RTL-SDR and GNU software radio program to demodulate and decode the data. Unfortunately, I do not have much experience in communications (I am only currently in Communications 1) so I might not have enough knowledge to successfully decode the digital signals going through the keyboards.
2. If I am not able to decode and recover the keyboard strokes, then I will make a video where I scan various frequencies to see what spectrum is there and what it looks like. For example tuning into wireless microphones (if available), keyboard transmission signal, LCD screen signal, RAM signal, etc. There are also many project where people use an SDR to tap into broadcast communications which were not originally intended for the public such as images from NOAA satellites, position data from airplanes and large marine vessels, etc. I will try to tap into these communications and see what I can create images of.

**Sources:**

Elliot, Melissa. "DEF CON 21 Presentation By Melissa Elliott - Noise Floor Exploring Unintentional Radio Emissions - YouTube." *Youtube.com*. Accessed October 31, 2013. <http://www.youtube.com/watch?v=5N1C3WB8c0o>.

"Rtl-sdr – OsmoSDR." *osmocomSDR*. Accessed October 31, 2013. <http://sdr.osmocom.org/trac/wiki/rtl-sdr#no1>.

Vuagnoux, Martin, and Sylvain Pasini. "Compromising Electromagnetic Emanations of Wired and Wireless Keyboards." EPFL. Accessed October 31, 2013. <http://infoscience.epfl.ch/record/140523/files/VP09.pdf>.



Nathan Tarrh  
COMP 116  
Final Project - Outline  
10/31/2013

Can you  
just  
focus  
on  
RFID?  
NFC  
is  
another  
beast

#### I. Abstract/Introduction

- A. RFID/NFC is a growing attack vector
- B. There are vulnerabilities all around us
  - 1. Charlie Card/Tufts ID
- C. Lots of literature around RFID hacking/cracking, less on hardware
- D. This project: designing a "hack-in-a-box"
  - 1. Goals
  - 2. Methods

#### II. Literature review/Prior work

- A. RFID Hacking - Live Free or RFID Hard - Francis Brown
  - 1. DEF CON 2013
- B. Exploring the NFC Attack Surface - Charlie Miller
  - 1. DEF CON 2012
- C. Anatomy of a Subway Hack - Russell Ryan, Zack Anderson, Alessandro Chelsea
  - 1. Restraining order from MBTA to stop presentation at DEF CON 16

#### III. Analysis/Project Design

- A. Given the prevalence of hacking (section II), where is the hardware?
  - 1. Focus of this project: "hack-in-a-box" similar to Pwn Plug
- A. Arduino-based design
  - 1. Arduino Uno
  - 2. Low-cost, low-power, small footprint
- B. RFID reader
  - 1. Store-bought vs. homemade(?) if possible
- C. Making the setup compact and portable
  - 1. Case, battery, etc.
- D. Code

#### IV. Results

- A. Success/failure of section III
- (B.) Using the hack-in-a-box to sniff Tufts IDs
- (C.) Using sniffed IDs to gain unauthorized access

#### V. Discussion

- A. Connect design to results
  - 1. What worked, what didn't
- B. Vulnerabilities revealed around campus
- C. Possible future work

## Node.js Vulnerabilities Outline

### Introduction

- What Node.js is
  - Explain what a framework is briefly
  - What other frameworks exist and who uses them
    - Django
      - <https://www.djangoproject.com/> (Instagram, Pinterest, Rdio, Mozilla)
    - Ruby on Rails
      - <http://rubyonrails.org/> (Basecamp, Groupon, Github)
  - Why Node is different
    - <http://nodejs.org/about/> (technical overview)
  - The pros and cons of those differences
- Why it is important
  - Node's usage is growing
    - Back this up
  - Big companies are already using it
    - <http://nodejs.org/industry/> (Uber, LinkedIn, ebay)
  - Mobile usage is rapidly increasing
    - Meaning more people will be looking to exploit web frameworks
      - Many mobile apps use frameworks
    - <http://www.forbes.com/sites/parmyolson/2012/12/04/5-eye-opening-stats-that-show-the-world-is-going-mobile/>

### Why did you choose this topic? Why is this important to know?

- As a developer interested in mobile development using a solid backend framework is very important
- As more people develop apps and websites it is important to be informed about what frameworks are secure
- Now, more than ever, it is important to secure user data
  - Edward Snowden
    - Make sure to secure as much data as possible
  - If companies rely on vulnerable software they could potentially be giving data away without knowing it
- There have already been serious flaws found in Node.js
  - Using eval("some\_string") in Javascript causes the string to be executed as Javascript
    - If exploited could give an attacker access to the entire server
    - Found: Ming Chow's Source 2013 Javascript talk (link to long)
  - Node did not properly check string length allowing attackers to get request headers or potentially spoof HTTP headers

## Security Implications of HTTP 2.0 - Outline

### Isobel Redelmeier - October 31

#### QUESTIONS SO FAR:

- How relevant is HTTP (1.1) these days?
  - Still gets used for things like handshakes and many pages
  - Is it being overtaken by SPDY and WebSockets (and other protocols)?
- Existing vulnerabilities in HTTP
  - ...and comparison with how HTTP 2.0 fixes them/doesn't change them/makes them worse
  - (Brief overview first; discuss throughout)
- HTTP's incorporation of SPDY
  - Built heavily upon SPDY
  - SPDY is used by Chrome, Firefox, and others... what about IE (eww)? Safari?
    - Microsoft wants HTTP S&M (hah)... \*Do they still?
- SPDY/HTTP 2.0 and encryption
  - BIG SUB-TOPIC!!!
  - Current issues with HTTPS today...
    - Improper implementation, NSA, key issues, etc.
  - Will encryption be blanketed/inherent?
    - Pros, cons - e.g., vs. performance
  - Microsoft's opposition...? *There is?*
  - How successfully can SSL actually encrypt multiplexed connections?!?!
    - \*Do much, much more research into this
- Security considerations listed in current specs  
(<http://http2.github.io/http2-spec/index.html#security>, last updated 10/27/13 - sub-headers here are straight from the specs):
  - Server authority and same-origin
    - Certificate verification... w/ multiplexing?
    - Longer lasting connections?
  - Cross-protocol attacks
    - "When using TLS, we believe that HTTP/2.0 introduces no new cross-protocol attacks. TLS encrypts the contents of all transmission (except the handshake itself), making it difficult for attackers to control the data which could be used in a cross-protocol attack. [[rfc.comment.5](#): Issue: This is no longer true]"
      - WTF?!?!?
      - Believed?!?! What part is "no longer true"?
      - WILL TLS always be used?
      - How does the handshake change in HTTP 2.0... and does this introduce new vulnerabilities here?
      - etc.
  - Intermediary encapsulation attacks

Hidden in Plain Sight: An Analysis of Private Browsing Methods  
Outline Due 10/31/2013  
Matt Brenman

Intro

- What does private mean?
  - Ambiguous (Could be private on many levels)
  - Misconception of users
- Why do people use/need privacy?
  - ACLU notes
  - Tor notes
  - No reason not to have it if they want it

What about  
Firefox?

Browser-supplied Private Modes

- What do they claim to do?
  - From Chrome Incognito mode: "You've gone incognito. Pages you view in this window won't appear in your browser history or search history, and they won't leave other traces, like cookies, on your computer after you close all open incognito windows. Any files you download or bookmarks you create will be preserved, however."
  - Warnings from Chrome:
    - Websites that collect or share information about you
    - Internet service providers or employers that track the pages you visit
    - Malicious software that tracks your keystrokes in exchange for free smileys
    - Surveillance by secret agents
    - People standing behind you
- Ways to break private browsing
  - From Target Computer
    - During Network Activity
      - External Application Methods
        - Parental Control Software (screenshots of web activity)
        - Key Loggers
      - Inner-Browser Methods
        - Add ons (Firefox) / Extensions (Google Chrome)
        - Default to blocked
  - After the Fact
    - Forensic Analysis (<http://www.mocktest.net/paper.pdf>)
    - Cookies in Flash Macromedia (Seems to be fixed)
  - From Computer on the Same Network
    - During Network Activity
      - Sniffing (Wireshark)
  - Other people who can track you

- ISP, netadmins (schools, businesses), the sites that you visit
- What's stopping them?
- Do not track header -- Nothing.

#### Other Methods of Private Browsing

- Simple IP Spoofing -- why it doesn't work
  - It doesn't send back to you!
- Proxy
  - Ask for info from proxy, proxy asks site for info, sends it back
  - You never technically asked the site for info, proxy did
  - Disadvantage
    - Looks like Javascript is disabled (?)
- VPN
  - Security more than Anonymity
- Tor (or general Onion Routing)
  - Advantages
    - Anonymity through routing
    - Encryption between relays
  - Disadvantages/Dangers
    - Browsing speed
    - Exit nodes remain unencrypted
      - Security issue -- consequences
      - Only as strong as the weakest (unencrypted link)

*Not Browser*

#### Conclusion

- Now that we know this, what should we do?
  - Awareness is most important, even if no method is ever used
  - If anonymity is needed or wanted, you can use any method listed (or not listed) above

#### Ideas for supporting material:

- Website where users can explore the different methods of private browsing and see examples (likely embedded videos or screenshots) of how to break them and (if applicable) how they work. Possibilities include wireshark screenshots comparing Incognito and standard browsing for Chrome, a step-by-step visualization of what each relay in Tor can see, and possibly a Track-Me section of the site where a user can see that websites do not have any responsibility to change their actions while a user uses private mode by having their information logged.
- Video backing up anonymity flaws

#### References:

- Analysis of Private Browsing:  
[https://www.usenix.org/legacy/events/sec10/tech/full\\_papers/Aggarwal.pdf](https://www.usenix.org/legacy/events/sec10/tech/full_papers/Aggarwal.pdf)
- Tor and the need for Anonymous Browsing:  
<https://www.torproject.org/>
- Info on VPNs:

<http://technet.microsoft.com/en-us/library/bb742566.aspx>

-Nothing to Hide Does Not Mean Nothing to Fear:

<https://www.aclu.org/blog/national-security/you-may-have-nothing-hide-you-still-have-something-fear>

-Needed: A paper on how to misuse Tor (unencrypted exit nodes)

Security Final Project Outline  
AJ Jenkins

Title: Broken Promises? An Analysis of Mobile Apps Promising User Privacy

Apps I might look into:

- Snapchat
- Whisper
- Ask.fm
- Spring.me
- Redphone
- TextSecure
- Kik messenger
- Voxer

Outline:

- Abstract
  - Already submitted
- Introduction
  - Which apps promise user privacy (e.g., Snapchat, Whisper)
  - Why privacy is desirable/valuable to users and why it's important that app developers deliver on their promises
  - Brief overview of some problems that have been found in apps like Snapchat
  - Also talk about the issue of cyber-bullying
- To the community
  - I became interested in this topic when I started using the app Whisper. It purportedly is an anonymous social network, but I know that's a hard thing to accomplish, so I was suspicious that it wasn't as secure as they made it sound.
  - This is an important topic because apps that promise user privacy provide a sense of security that leads to people sharing more personal things than they usually would. Although the average person might not care about their recipes being intercepted by a third party, the data users share with these applications are typically more sensitive. Security is a hard problem, and many app developers are small teams of people and can't protect users' privacy like they're promising.
- Action Items (these are just ideas, I'm not sure what can be done yet)
  - How to save images in Snapchat without the other user getting a screen-capture notice (an app on the app store does this)
  - How to find the hidden image files that Snapchat saves on your phone (this is possible, but supposedly challenging)
  - Intercepting Snapchat photos sent over a network (I don't know if this is possible — they might be encrypted)

Email me to  
send you  
Snapchat  
Forensics  
article

TITLE:

Security Flaws in Cryptographically Secure Pseudorandom Number Generators

CONTENTS:

- 1 Introduction
  - 1.1 Pseudorandom Number Generators
  - 1.2 The Bad Guys
  - 1.3 To The Community

← what about  
SPRNGs?

- 2 Breaking the PRNG

- 2.1 Cryptocat (Improper implementation)

- <http://nakedsecurity.sophos.com/2013/07/09/anatomy-of-a-pseudorandom-number-generator-visualising-cryptocats-buggy-prng/>

- 2.2 Dual\_EC\_DRBG (Backdoor)

- <http://blog.cryptographyengineering.com/2013/09/the-many-flaws-of-dualecdrbg.html>

- 2.3 /dev/random (Insufficient Entropy Accumulation)

- [https://www.schneier.com/blog/archives/2013/10/insecurities\\_in.html](https://www.schneier.com/blog/archives/2013/10/insecurities_in.html)

- 3 Prevention

- 4 Conclusion

SUPPORTING MATERIALS IDEA:

Replication of Sophos test. Code that captures the results of multiple runs of Cryptocat PRNG and another PRNG. Results will show that a certain result will be twice as likely in the Cryptocat version.

-> Cryptocat blog discussing the vulnerability:

<https://blog.crypto.cat/2013/07/new-critical-vulnerability-in-cryptocat-details/>

-> Github pull request fixing bug:

<https://github.com/cryptocat/cryptocat/commit/5c69cb7d8543184d2f33944cb4129605d050ecad>

-> Sophos example:

<http://nakedsecurity.sophos.com/2013/07/09/anatomy-of-a-pseudorandom-number-generator-visualising-cryptocats-buggy-prng/>



Denis Richard  
High-Level Outline  
BGP Spoofing

Intro

- BGP as an interdomain communication tool – how it works
- Examples of problems previously experienced with BGP
- Quick overview of its inherent vulnerabilities
- Quick overview of potential remedies

Why is it a problem?

- Interception
- Black holes
- Redirects

What are the vulnerabilities?

- Trust based system with no supreme oversight

Why is this problem so hard to fix?

- Pairwise keying is expensive -  $O(n^2)$
- Solutions require all ISPs and ASes to participate

What are potential solutions?

- Secure BGP
- Pairwise keying

Conclusion

---

Supplemental material idea

- A video showing a visual demonstration of how BGP works and how it can be exploited

References

- <http://arxiv.org/pdf/1205.4564v1.pdf>
- <http://ix.cs.uoregon.edu/~butler/pubs/bgpsurvey.pdf>
- [http://www.nanog.org/meetings/nanog49/presentations/Tuesday/HowSecure\\_NANOG\\_print.pdf](http://www.nanog.org/meetings/nanog49/presentations/Tuesday/HowSecure_NANOG_print.pdf)
- <http://dl.acm.org/citation.cfm?id=2382254>
- <http://www.blackhat.com/presentations/bh-europe-03/bh-europe-03-dugan.pdf>
- [http://www.cs.jhu.edu/~fabian/courses/CS600.424/course\\_papers/sbgp.pdf](http://www.cs.jhu.edu/~fabian/courses/CS600.424/course_papers/sbgp.pdf)
- <http://www.ir.bbn.com/sbgp/>
- [http://www.cisco.com/web/about/ac123/ac147/archived\\_issues/ipj\\_6-3/securing\\_bgp\\_s-bgp.html](http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_6-3/securing_bgp_s-bgp.html)
- <https://www.defcon.org/images/defcon-16/dc16-presentations/defcon-16-pilosov-kapela.pdf>
- <http://www.youtube.com/watch?v=IDoivWHVrGI>

*I think this was how L0ph0t could bring down the Internet.*  
*→ please explain*

## Comp 116 Final Project Outline

- I. Introduction to Industrial Control Systems (ICS), including SCADA
  - A. Example applications
    - i. Electrical grids and generating stations
    - ii. Water management
    - iii. Building management systems (BMS)
  - B. Ubiquity – 90 millions SCADA
  - C. ICS increasingly connected to internet
  - D. Public cyber security incidents
    - i. Stuxnet (source code on GitHub)
    - ii. Flame
    - iii. Lesser events
- II. The LonWorks Platform
  - A. Hardware description
    - i. Goals
    - ii. Comparison to Ethernet
  - B. Standard protocols
    - i. International
      - a) “Control Networks” defined by ISO/IEC 14908.1 through 14908.4
      - b) “Control Networks” published as ISO/IEC JTC 1/SC 6
    - ii. United States of America
      - a) ANSI/CEA 709.1-C-2010 – Control Networking Protocol Specification
      - b) IEEE 1473-2010 – Communications Protocol Aboard Passenger Trains
      - c) NIST 800-82, “Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control System Security”
    - iii. China – GB/Z 20177.1-2006 – Control networking and building controls
  - C. Protocol comparisons
    - i. Similarities between Ethernet-TCP-IP and LonWorks
    - ii. Differences between Ethernet-TCP-IP and LonWorks
  - D. Increasing use of TCP-IP with LonWorks hardware
- III. ICS Vulnerabilities
  - A. Explosion of know issues
    - i. 15 public alerts in 2010
    - ii. 129 public vulnerabilities in 2011. Most of the discoveries were made by Luigi Auriemma, a hobbyist researcher in Malta.
  - B. Compare and contrast with common internet-based attacks
    - i. Buffer overflow
    - ii. Man in the middle
    - iii. Denial of service
    - iv. Backdoors
    - v. SQL injection
- IV. Security Best Practices

Don't  
make  
this  
too  
long

The project is the visualization of PCAP files captured by the instructor at DEF CON. It provides for anomaly detection against statistical trends, as well as high-level summaries, of potentially millions of packets. Given the dragnet surveillance programs undertaken by both the U.S. government and commercial “data vendors”, packet analysis tools that scale are becoming critical.

## 1 Section Outline

- Abstract
- Introduction
- To the Community - both the security and viz communities
- Prior Work - an informal draft follows
- Applications - a tour of the software
- Implementation Notes - how I made it performant, and aesthetic decisions
- Summary
- References

why do I care?

## 2 Prior Work

Existing security visualizations primarily focus on individual packets or other binary data files, such as detecting anomalously long ASCII strings in mp3 files. Greg Conti<sup>1</sup> created rumint,<sup>2</sup> a package that allows for these sorts of visualizations to be created easily. Rumint’s binary rainfall allows the user to see (literally) the similarities and differences among packets. Conti also deploys standard multivariate plots to chart arbitrary numeric data.

The visualization to be created for this project, though less general and comprehensive than rumint, improves upon Conti’s work in the following ways:

1. By tolerating orders of magnitude more packets than rumint by the use of binning in 15-second<sup>3</sup> intervals and extensive preprocessing. It is hoped that trends over hours or days will emerge.

---

<sup>1</sup>Security Data Visualization, 2012

<sup>2</sup>rumint.org

<sup>3</sup>Subject to change.

2. By focusing on heterogenous packet data, searching for statistical trends and anomalies, rather than bit changes across mere dozens of packets from a single stream.<sup>4</sup>
3. By using visualizations tailored to the data, e.g. a world map for GeoIP information, rather than a scale of longitude values.
4. By allowing the interactive selection and query of packets, filtered across multiple dimensions.
5. By being more portable than rumint, achieved by running in the browser rather than on Microsoft Windows.

### 3 Supporting Material

A fully-working visualization will be produced, based on the “coordinated multi-view” model where multiple visualizations work in tandem. Time and technical abilities permitting, the primary view will be a scatterplot or line plot of numeric data over time, such as unique hosts, number of packets, download sizes, and so on. Also present is a world map of GeoIPs, additional min/max information, and some visualization of the IP address blocks. Selecting data in the main timeline will recompute the other views on only that subset.

I have already prepared a preprocessor in Ruby using the PacketFu and GeoIP gems (geoIPs have been found for 99.99% of remote hosts in preliminary runs). It includes logging capabilities and requires a range of packet numbers to be specified at the command line. This will allow multiple instances to run in parallel on a high-performance machine,<sup>5</sup> and then be concatenated together. Currently I expect binning (combining packets into 15-second intervals) to happen as a second pass, with the first pass not combining packets in any way, merely extracting relevant data from the pcap files.

Once the final csv is prepared, visualization will be done by d3.js. Performance degrades with high numbers of DOM elements, so it is hoped that preprocessing and potentially prerendering will allow the performant visualization of millions of packets. As a fallback, I may generate a static svg image.

---

<sup>4</sup>This is not to say one focus is superior than the other, merely different.

<sup>5</sup>Wormhole.

Paul Pemberton

10/31/13

Comp 116

## Final Project Outline

### Paper Outline:

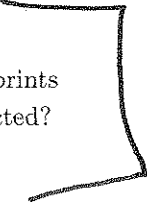
0. Abstract: See already turned in.
1. Introduction (what is the issue presented)
  - a. Internet browsers are equipped with script blocking and script warning mechanisms
  - b. No system is commonly available that allows for users to see real-time tracking of what scripts are doing
  - c. Binary treatment of scripts (OK or malicious) is not good enough
2. To the community
  - a. In the same way that virus software is used to actively monitor a user's computer, there is a need for software that will help users avoid website script based attacks
  - b. Currently existing software is not good enough for the community for the following reasons:
    - i. The systems that are easily accessible (in the form of extensions to browsers) do not effectively protect users from threats
    - ii. A binary system (i.e. sites are either good or bad) does not provide enough information to the users
    - iii. Expired certificates are a constant problem and open users to social engineering attacks (if someone is used to accepting an expired certificate, it will be easier to trick them into entering a site with an expired certificate)
    - iv. Live script monitoring will allow users to see real time script activity, and thus users will have a spectrum of warnings and associated risks to be informed about
3. Defenses
  - a. The goal of the defense portion of this is to allow for real-time calculation and classification of scripts that are running on a web site
  - b. Currently, the common analysis done on scripts that are run on websites is a static style analysis.
    - i. Static vs. dynamic analysis of malicious code/scripts
    - ii. Discuss issues pros and cons of both
  - c. Dynamic analysis and real-time display of risk will allow users to better avoid both script based attacks as well as script based social engineering attacks
  - d. Program description:
    - i. Analyze scripts and tag malicious pieces (certain keywords)
    - ii. Score scripts based on malicious scoring matrix
    - iii. Determine overall risk of script and display combined score as a risk factor on browser (display to user via extension)
4. Conclusion
  - a. Discuss why this dynamic script analysis and real-time risk application is necessary
    - i. Binary is not good enough
    - ii. Danger of social engineering attacks
  - b. Discuss methods for dynamic analysis of website scripts
  - c. Discuss risk factor calculations

NoScript

JavaScript  
Static  
dynamic  
analysis  
Pick  
Sam  
Guyer's  
brain

## 1 Outline

1. Abstract
2. Introduction - Web Applications as they are now
  - (a) What is a WAF?
3. To Whom This May Concern
  - (a) Audience
    - i. Anyone who considers the use of a WAF
      - A. Buisness
      - B. Those who host others' unsafe web apps
    - ii. Developers of web applications
  - (b) Motivation
    - i. Should this not be safe, WAF becomes a fancy coating of code on top of web app code. In other words, useless.
4. Action Items
  - (a) The Good - What they (claim to) do well
    - i. Blocking known attacks.
  - (b) The Bad - What fails
    - i. The problem with a black list / fingerprints
    - ii. What happens when an attack is detected?
    - iii. How ugly can it get?
  - (c) Defenses and Lessons
    - i. Complete removal of WAFs
5. Conclusion



This is  
the  
meat &  
potatoes  
of  
your  
paper

## 2 Supporting Material

- Break through a default web app firewall into an insecure location (CTF VM)
  - The idea is that it doesn't matter if (a) the firewall is up or (b) you'll fix it later, I only need to break in once to do ANY damage. If you're not careful on the app side, you'll regret it.

*Outline*

## Abstract for Network Security Final Project

Jacob Schmitz

My project is an attempt to create a demonstration of a reproducible exploit of an Apache Hadoop Distributed File System (HDFS). There are several tidbits online about security vulnerabilities, such as: <http://www.cvedetails.com/cve/CVE-2012-3376/> but none provide a clearcut scenario for when an exploit can be exploited.

The motivation behind this is twofold. One is to point out a security flaw in the industry standard distributed file system, but also to raise awareness about how many major companies and government agencies (maybe not a surprise) store incredibly sensitive information about hundreds of millions to people. Some of the organizations in the industry that use Hadoop are enumerated in a Black Hat presentation I dug up that discusses some of the security vulnerabilities that were known a few years ago. They included Facebook, Amazon, Yahoo, the NSA, LinkedIn, IBM, and many others. The kind of data these organizations store in a Hadoop cluster covers just about everything you can imagine (data on the petabyte scale). For financial institutions, this stuff is actually critically important to processing and validate the millions of transactions that occur every day. These are just the big groups, but easier and easier to install and managing Hadoop clusters via prepackaged distributions like Cloudera's Distribution of Hadoop (CDH) and software to manage clusters (Like Cloudera Manager) and monitor the health of individual nodes. These tools are free for basic versions, which can actually allow users to do some very powerful things, and these aren't necessarily huge companies with dedicated security teams. These could be individual people even. The result is a high risk that this relatively new technology will be deployed insecurely.

I chose as a mentor Alan Jackoway, a Software Developer for Cloudera's Customer Operations Tools Team (basically builds internal tools using Hadoop and related technologies). I worked with Alan as an intern this summer and he mentored me on my project and he agreed to help me with this. His impression is that the first step towards a proof of concept, even before setting up Kerberos security and

# Comp116 Final Project Outline

Dave Mancinelli

October 31, 2013

## Abstract

Public WiFi hotspots have become ubiquitous in coffee shops, airports, and other commercial venues where consumers have come to expect internet connectivity for their laptops, smart phones, and tablets. The traffic on public networks is largely unencrypted, but public WiFi users, many of whom are unaware of the associated risks, continue to log in to email accounts, Facebook, bank accounts, or any other domains containing sensitive personal information. Such an environment – wherein a large base of users sends personal data across an unprotected network – could also be a hotspot for nefarious hacker activities. This paper examines the known hacker exploits of public WiFi networks, such as man-in-the-middle attacks, packet sniffing, and ARP spoofing; analyzes the risks of using public networks; and provides recommendations for safer public WiFi usage.

## 1 Introduction

Self explanatory...

Outline

## 2 Privacy and Data Leakage

Summary of hotspot prevalence, types of data loss, discussion of privacy...



### 3 Attack Vectors

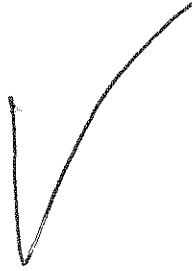
This section describes several attack strategies that can be used on open WiFi networks...

Sniffing

Man-In-The-Middle

Rogue Access Points

Denial of Service

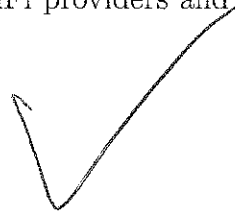


### 4 Protection Practices and Strategies

This section details steps that public WiFi providers and users can take to mitigate attacks...

Providers

Users



### 5 Supporting Material

**Methodology** Supporting material will include an actual demonstration of sniffing and MITM attacks on open WiFi hotspots, such as Starbucks. This will include an analysis of sniffed packets as an empirical study of the types of data that can be intercepted over a public network.

Data Analysis

### 6 Conclusions

Self explanatory...

Please  
use a  
good  
antenna.  
(e.g. alfa)

## References

- [1] R. G. Brody, K. Gonzales, and D. Oldham. Wi-fi hotspots: secure or ripe for fraud? *Journal of Forensic Investigative Accounting*, 5(2):27–47, december 2013.
  - [2] A. K. Elliot. User’s perception about security of the public wireless network. *International Journal of Societal Applications of Computer Science*, 2(8):434–438, august 2013.
  - [3] T. Kindberg, C. Bevan, E. O’Neill, J. Mitchell, J. Grimmett, and D. Woodgate. Authenticating ubiquitous services: a study of wireless hotspot access. In *Proceedings of the 11th international conference on Ubiquitous computing*, Ubicomp ’09, pages 115–124, New York, NY, USA, 2009. ACM.
  - [4] A. Matos, D. Romão, and P. Trezentos. Secure hotspot authentication through a near field communication side-channel. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on*, pages 807–814, 2012.
  - [5] S. Y. Nam, S. Jurayev, S. S. Kim, K. Choi, and G. S. Choi. Mitigating arp poisoning-based man-in-the-middle attacks in wired or wireless lan. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):1–17, 2012.
  - [6] M. M. Noor and W. H. Hassan. Current threats of wireless networks. In *The Third International Conference on Digital Information Processing and Communications*, pages 704–713, 2013.
  - [7] K. Singh, H. J. Wang, A. Moshchuk, C. Jackson, and W. Lee. Practical end-to-end web content integrity. In *Proceedings of the 21st international conference on World Wide Web*, WWW ’12, pages 659–668, New York, NY, USA, 2012. ACM.
  - [8] J. Spaulding, A. Krauss, and A. Srinivasan. Exploring an open wifi detection vulnerability as a malware attack vector on ios devices. In *Malicious and Unwanted Software (MALWARE), 2012 7th International Conference on*, pages 87–93, 2012.
- [2, 1, 3, 4, 5, 6, 7, 8]

# Outline

## Visual Geolocation Based Black and White Listing

Karl Cronburg\*

*Dept. of Computer Science, Tufts University, Medford MA 02155*

### Abstract

Various commercial and open-source tools are available for managing IP-based filtering of web traffic. A common open-source method is the use of static ‘tricks’ found on stackoverflow (e.g. defining Apache rewrite rules). A more involved approach is to create *iptables* rules. However, the availability of tools to interact with iptables from a high level is lacking. As such, we present a novel approach to managing iptables rules through an interactive map of the world powered by Google Maps and IP2Location.com. Our approach focuses on effectively conveying web site traffic in a way that any content-provider can understand, regardless of prior experience with tools like iptables. We discuss the effectiveness of existing black and whitelisting tools, and where our approach fits into today’s network security landscape. We conclude with a discussion of possible improvements to our approach, which should lead to increased productivity of website administrators in dealing with network attacks.

---

\*Electronic address: `karl@cs.tufts.edu`

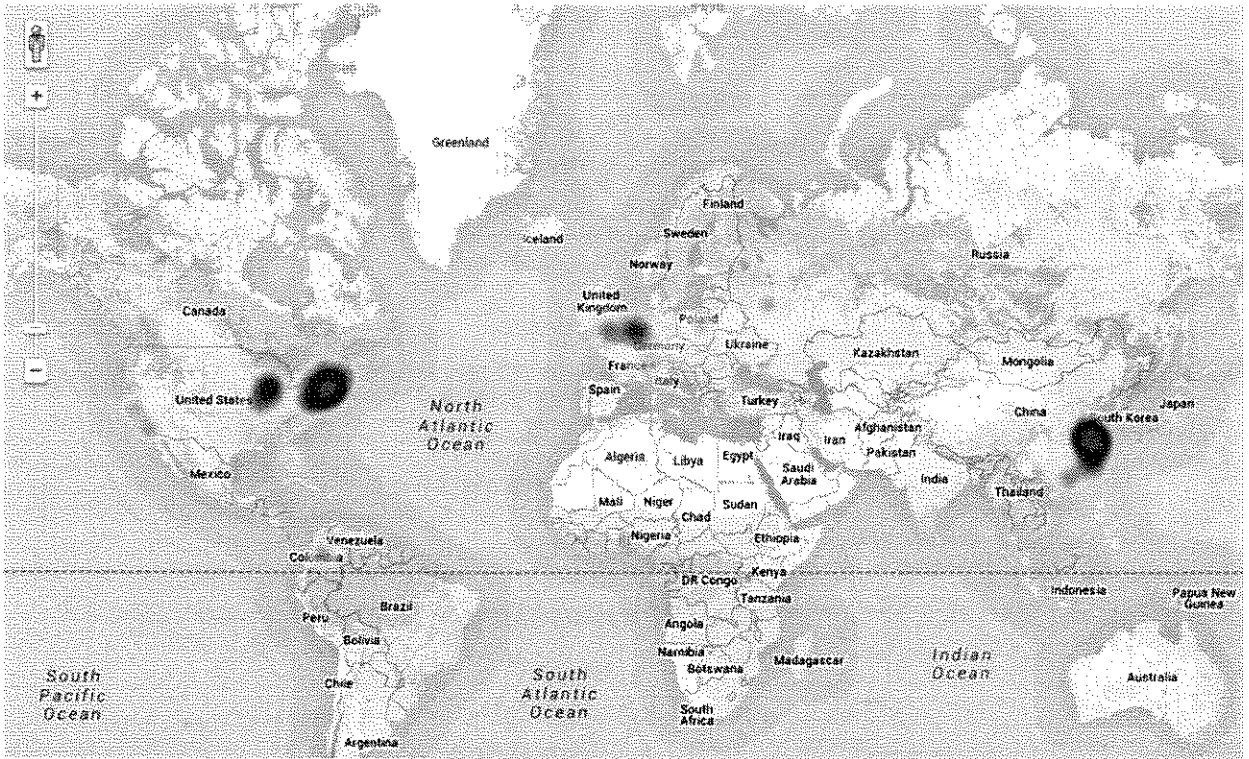


FIG. 1: A heatmap of static / historical data from a personal web-server open to the Internet for the past few years. The data was manually filtered to show only suspicious activity, based on known legitimate usage of the server. The data may however contain a number of false positives located in Pennsylvania and Boston, the two primary locations the server was used.

#### D. Blacklisting and Whitelisting

1. ‘Teaching’ syntax for `iptables` using a separate HTML frame to contain a list of rules automatically generated based on what the user does to the map.

## IV. APPLICATIONS

### A. Website Administration

### B. Big Data

What could  
possibly  
go  
wrong?

## V. FUTURE WORK

1. Possibly show live packet statistics and / or live graphs to help the user correlate large traffic bursts with packet and data types.
2. Allow advanced users to edit `iptables` rules themselves. Would need to do syntax verification and input cleansing to make sure only `iptables` rules are performed.
3. Add a color-bar to indicate the exact rate at which requests are originating from a given location.
4. A hover textbox feature for telling the user the exact rate at which requests are originating from the location over which the mouse is hovering.

## VI. CONCLUSIONS

- 
- [1] J. Zhang, A. Chivukula, M. Bailey, M. Karir, and M. Liu, in *Proceedings of the 14th international conference on Passive and Active Measurement* (Springer-Verlag, Berlin, Heidelberg, 2013), PAM'13, pp. 218–228, ISBN 978-3-642-36515-7, URL [http://dx.doi.org/10.1007/978-3-642-36516-4\\_22](http://dx.doi.org/10.1007/978-3-642-36516-4_22).
  - [2] J. Wu, P. Teregowda, J. P. F. Ramírez, P. Mitra, S. Zheng, and C. L. Giles, in *Proceedings of the 3rd Annual ACM Web Science Conference* (ACM, New York, NY, USA, 2012), WebSci '12, pp. 340–343, ISBN 978-1-4503-1228-8, URL <http://doi.acm.org/10.1145/2380718.2380762>.
  - [3] A. G. West and I. Lee, in *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference* (ACM, New York, NY, USA, 2011), CEAS '11, pp. 73–82, ISBN 978-1-4503-0788-8, URL <http://doi.acm.org/10.1145/2030376.2030385>.

Ashley Hedberg  
Comp 116  
10/31/2013

## Final Project Proposal: The "Privacy" of Private Browsing

### Outline:

#### I. Introduction

##### A. Overview of private browsing options on major browsers

1. Google Chrome – incognito mode
2. Mozilla Firefox – private browsing
3. Internet Explorer – InPrivate browsing

##### B. What private browsing options claim to keep private

1. Browsing history
2. Temporary internet files
3. Usernames and passwords
4. Form data
5. Cookies

##### C. Known issues

1. Artifacts left in virtual and browser memory
2. Pagefile on Windows machines
3. Does not provide anonymity to ISP or websites
  1. Private browsing does not, in general, claim to do this
  2. Highlights difference in motivations between users who want to keep their browsing sessions private and the browser developers

#### II. Applications<sup>1</sup>

##### A. Reconstruction of web browsing by digital forensics professionals

##### B. Exploited by developers of malware

##### C. Highlights need for a dialogue between users and browser developers about what private browsing should be

#### III. To the Community

##### A. Your private browsing session is not hiding as much information as you might think

##### B. If you have a vested interest in keeping your information private, find an alternate method

<sup>1</sup> The content of this section will be largely determined by the supporting material that I submit with this paper. My goal is to devise a program that will take various known flaws of private browsing and use them in combination to attempt to reconstruct a user's browser history. If this is successful, I would discuss the potential uses and abuses of this program under the applications section.

May also want to add in Tor Browser

Here's an idea: perhaps use Salite go through what's stored on machine by Chrome PR

## When Analytics Cross the Line

### Outline

#### Introduction

- What data mining is
- How data mining works (screen scrapes, publicly available data, mobile devices)
- Source (Introduction to Data Mining): <http://www.twocrows.com/intro-dm.pdf>

#### Data Mining Examples

- Detecting fraud risk for the banking industry
- Facebook graph search (social interest)
- Source (JPMorgan Uses Data Mining to Detect Fraud): <http://finance.yahoo.com/news/jpmorgan-mining-big-data-detect-121150702.html>
- Source (Facebook Graph Search for Data Mining): <http://www.pcworld.com/article/2056080/facebook-stalker-tool-uses-graph-search-for-powerful-data-mining.html>

#### Value of Information

- Data mining makes money by compiling market data and selling it
- Data Brokers and how they work (ex. Acxiom)
- Example of what can be inferred from basic information (see: Supporting Material)
- Source (Acxiom Privacy Policy): <http://acxiom.com/about-acxiom/privacy/us-products-full-privacy-policy/>

#### Ethical Implications

- Why data mining is an infringement on basic human rights
- The data can be wrong and you are unable to either fix it or unable to get it removed (data brokers have a ridiculous amount of power)
- General intention of legal privacy is not being upheld (Electronic Communications Privacy Act is outdated and, as a result, unconstitutional)
- Source (ACLU on Internet Privacy): <https://www.aclu.org/technology-and-liberty/internet-privacy>
- Source (Data Brokers with Wrong Information): <http://144.142.224.168/news/wxin-data-brokers-wrong-information-093009,0,911967.story>

#### Actions Items

- Demand your dotRights: <https://www.dotrights.org/>
- American Civil Liberties Union: <https://www.aclu.org>

Have you  
You  
should  
RUN  
to  
see  
Prof.  
Remco  
Chang

## Overview of Cooperative Infrastructure Defense and Ant Based Cyber Defense

Hayley Weiss

[Hayley.Weiss@tufts.edu](mailto:Hayley.Weiss@tufts.edu)

Ming Chow

### Abstract:

Today we monitor cyber defense by gathering data across an infrastructure to a single point and analyzing it centrally, which is problematic as it scales poorly. To combat this, the Pacific Northwest National Laboratory (PNNL) has been working on a method called Cooperative Infrastructure Defense (CID), which utilizes “digital ants” and “swarming intelligence” to quickly react and adapt to cyber attacks with humans supervising at the appropriate level. This paper gives an overview of the concepts behind CID and digital ants.

### Introduction:

#### The Community:

- Talk about the problems with current defense, scoping etc.

- Concerted cyberdefensive actions spanning organizational boundaries is difficult

- Cyber adversaries not hindered by central coordination

- Talk about the pros of CID and ABCD

- Rapid and automatically adapt to new attacks

- Enables humans to supervise at appropriate level

- Software agents share decision making power and handle realtime portion

- Scalable, dynamic, robust framework for securing complex computational infrastructures

- Talk about what has been done so far

- Have simulations

- Prototypes for UI

- Prototypes for mobile sensor agents

#### Discuss the CID hierarchy and ants

- Supervisor

- Humans

- Provide guidance and receive feedback

- Only do things when contacted by lower level agent

- Or when take initiative, but is discouraged

- Sergeants

- Top level agent of an enclave

- Situational awareness to supervisor via visualization interface

- Translates guidance into actionable policy



## Sentinel

- mid level rational software agents
- responsible for single machine or groups of machines
- interface with the sensors
- provide geography to sensors and mobility, as well as rewards and

## spawning capabilities

- give feedback
- talk about a few of the functions

## sensor

- lowest level
- lightweight software agents
- talk about ANTS here
- modeled after insets
- simplistic logic
- controlled by sentinels
- talk about implementation as IP packests

*Does this actually work?*

## Conclusion/summary

- Reiterate what CID and ants are
- Why useful
- Etc.

## Supporting materials ideas:

- Create an ant packet?
- Implement some of the simple sentinel functions?

## References:

Bland, Eric. "Digital 'ants' take on computer worms." 28 Oct. 2009. [online].  
Available: <http://www.nbcnews.com/id/33509921>

"DigitalAnts™: Ant-Based Cyber Defense" Pacific Northwest National Laboratory.  
[online] Available: <http://i4.pnnl.gov/news/digitalants.stm>

King, Anna. "New Model for Cyber Security: Ants." 2 Aug. 2011. [online]  
Available: <http://www.npr.org/templates/story/story.php?storyId=138941257>

"Ants vs. worms: New computer security mimics nature." 25 Sep. 2009. [online]  
Available: <http://phys.org/news173108776.html>

J. Haack, G. Fink, et. Al. "Ant-Based Cyber Security"  
[http://hivemind.cs.ucdavis.edu/\\_docs\\_/ITNG-290-Ant-Based-Cyber-Security.pdf](http://hivemind.cs.ucdavis.edu/_docs_/ITNG-290-Ant-Based-Cyber-Security.pdf)

Fink, Glenn, Oehmen, Christopher. "Final Report for Bio-Inspired Approaches to Moving-Target Defense Strategies." 30 Sept. 2012

1. Intro
  - a. Assumptions
    - i. Using modern languages/frameworks
    - ii. Typically automatically cover SQL injections/ XSS. They're boring
  - b. Balancing hardening and development
  - c. Overview of consequences of never doing security hardening
  - d. What's good enough?
  - e. Overlooked problems
    - i. route logic
    - ii. pw safety
    - iii. old software
    - iv. more tbd
2. Consequences
  - a. It gets harder later
  - b. Open to more attacks when you go live
  - c. Hurts investors opinions of you if they find out about the attack
3. Good enough?
  - a. When to start?
    - i. Just before going live
    - ii. With few/no users, you have no information for a hacker to bother breaking. You're just not a target
  - b. How much?
    - i. Until you or your team cannot think of more ways to break the system.
      1. You know the system best but get someone who isn't as familiar to give a fresh view
    - ii. What does a break mean for you?
      1. Don't have lot's of private data: breaks aren't a huge priority
      2. Certain modules may be more important than others to harden
        - a. Harden your user password DB
4. Overlooked problems
  - a. Route logic
    - i. Reset anyone's pw
    - ii. database dumps with smart queries (tuftstext, nomic)
  - b. PW safety
    - i. sent ITC (in the clear):  
<http://techcrunch.com/2010/11/18/yet-another-hot-startup-leaves-a-gaping-security-hole-in-its-iphone-app/>
    - ii. sent ITC: Nomic
  - c. Old software
    - i. Rails breaks daily
    - ii. apache/nginx occasionally have holes
    - iii. updating isn't on your mind

What about  
training  
developers?  
Think  
of  
abuse  
cases.

Patch +  
band aids  
don't  
work

## Digital Rights Management Outline

- Copy protection technologies
  - CD/DVD fingerprinting AKA Data Position Measurement
    - ex: SecuROM
    - encrypt part of the CD/DVD and encode the key to decrypt it within the format of the disc itself
    - method for secuROM
      - either send your master CD and let them encrypt and replicate it
      - or encrypt it with their Online Encryption Toolkit (OETK) and send it to a licensed replication plant
    - controversy
      - installs device drivers to your computer without your knowledge
    - workarounds/fixes
      - use an image burning software with data position measurement
      - cracked executable
  - online activations
    - ex: SecuROM again
    - force user to authenticate their particular disk with an online server, once again using the same sort of fingerprinting/key storage as in Data Position Measurement
    - can be used without a CD or DVD
      - useful for electronically distributed games
    - limited number of activations
      - can also set expiration date for activation, for use in free trials
    - controversy
      - can only use software a limited number of times, after which it becomes useless
      - reinstallation counts as an activation
      - Installed without prior knowledge or consent (see Spore lawsuit)
    - workarounds/fixes
      - Began offering deactivation so you can reuse activations
      - cracked executable
  - always online
    - ex: Origin, Steam
    - require constant, or near constant, internet connection in order to use any software
    - controversy
      - can't play the game if you don't have internet
      - relying on the integrity of third party servers in order to use your content
      - if the company eventually shuts down servers your game is

entirely worthless

- workarounds/fixes
  - cracked executable, specific dll files for steam
  - for something like SimCity (Origin), it's necessary to create a fake authentication server that is always running in the background
- Warez groups
  - Examples: RELOADED, Razor 1911, SKIDROW
  - "The Scene"
    - Groups that compete to release cracked or stolen video games, films, television shows, music, etc...
    - Culture grew up around reverse engineering software
      - compete against each other, have strict rules for releases
  - Cracked executables
    - release versions of the game executable as well as various dynamic link libraries and other game data that are free of DRM
  - Intros, demos, etc...
    - Legitimate portion of the Warez groups that encourages art combined with programming skill
  - Distribution
    - cracks themselves can be directly downloaded from a site like the skidrow site
    - full games+cracks are almost exclusively distributed through torrents because of the decentralized aspect and lack of responsibility
- Consumer-cracker-software company relationship
  - cracker-software company
    - arms race
    - warez groups take pride in shaming software companies
    - DRM companies market their software as "Hard on hackers, easy on you"
  - cracker-consumer
    - provide illegitimate method for a user to obtain software
    - provide a method for legitimate customers to use software broken by over protective DRM (see the Ubisoft using a RELOADED crack source)
  - consumer-software company
    - some companies have gone for more and more DRM like Ubisoft and EA
      - Spore and Simcity are both great examples of this
      - Xbox One tried to do this... and failed miserably after incredible backlash
        - <http://news.xbox.com/2013/06/update>
    - others market their lack of DRM as a selling point
      - [http://www.gog.com/news/2013\\_nodrm\\_summer\\_sale\\_begins\\_on\\_gogcom](http://www.gog.com/news/2013_nodrm_summer_sale_begins_on_gogcom)
      - <http://cdpred.com/no-drm-in-the-witcher-3-wild-hunt-an-open-letter-to-the-community/>

Sources:

Creating an image of a SecuROM protected DVD with Data Position Measurement:

<http://support.alcohol-soft.com/knowledgebase.php?postid=24649&title=Creating+an+Image+of+a+Securom+protected+DVD>

Ridiculous roundabout solution to SimCity's always online DRM:

<http://patheticreviews.com/2013/06/12/how-to-install-simcity-2013-bypass-drmplay-offline/>

Describes Data Position Measurement:

<https://www.google.com/patents/US7463565>

Describes SecuROM disc-based protection:

[https://www2.securom.com/fileadmin/user\\_upload/downloads/SecuROM\\_Disc.pdf](https://www2.securom.com/fileadmin/user_upload/downloads/SecuROM_Disc.pdf)

Describes SecuROM product activation:

[https://www2.securom.com/fileadmin/user\\_upload/downloads/SecuROM\\_Product\\_Activation.pdf](https://www2.securom.com/fileadmin/user_upload/downloads/SecuROM_Product_Activation.pdf)

Ubisoft support using a RELOADED crack:

<http://arstechnica.com/gaming/2008/07/ubisoft-drm-snafu-reminds-us-whats-wrong-with-pc-gaming/>

Class action lawsuit against EA for using SecuROM:

<http://www.courthousenews.com/2008/09/23/Spore.pdf>

Information on "The Scene" and their lingo from a popular cracking group:

<http://skidrowcrack.com/f-a-q/>

More information on another cracking group, one of the oldest:

<http://www.razor1911.com/demo/?menu=history>

I've sent several emails/phone calls to various people in Sony DADC (owner of SecuROM) looking for an interview to get more information. We'll see if I get a response though

I'm also going to try to get in touch with the warez groups themselves to see if anyone wants to work through a cracked executable with me

Idea for supporting materials:

- Walkthrough of how a cracked executable actually works
- Source code of a crack with comments describing what changes were made

# Time-Lock Cryptography: Outline

William Clarkson  
william.clarkson@tufts.edu

October 31, 2013

Have you  
talked  
to  
Ben  
Hegert  
about  
this?

## 1 Introduction

## 2 Applications

I will discuss in detail some of the most important applications of time-lock cryptography, particularly how it will solve problems which currently cannot be solved without relying on a trusted third party.

## 3 Limitations

I will discuss what is actually currently possible with time-lock cryptography, specifically the fact that puzzles can only be devised to withstand a certain amount of sequential computation, not an actual amount of wall clock time.

## 4 The Basic Algorithm

I will describe the simple form of a time-lock cryptography algorithm.

## 5 Improvements to the Algorithm

I will discuss clever ways that the creation of the time-lock puzzle can be parallelized so a large amount of computation will be required to unlock a puzzle which can be created in a much shorter period of time.

## 6 Implementation (Supporting Material)

I will briefly describe the details of my example implementation of a time-lock cryptography puzzle, and any interesting discoveries that I made while developing it.

## 7 Conclusion

## References

- [1] R. Rivest, A. Shamir, D. Wagner. "Time-lock puzzles and time-release Crypto. March 10, 1996.
- [2] Time-Lock Encryption. <http://www.gwern.net/Self-decrypting>

## The Downfall of Silk Road

1. Some History
  - a) Silk Road, online drug retailer
  - b) Bitcoin, anonymous transactions
  - c) The Farmer's Market: Silk Road's predecessor
    - How did it get taken down?
    - Subpoena- owners used Hushmail, FBI able to find servers
2. The downfall of Silk Road
  - a) Ulbricht's big mistake → personally identifiable information on the internet
  - b) Ulbricht needed help with the technical aspects of the site
    - Looked for help on a forum called "Bitcoin Talk"
    - Searched for "IT pro in the Bitcoin community"
  - c) "Altoid" handle- linked with his real gmail, including first and last name
  - d) Gmail address + Silk Road Source Code allowed tracking of IP to a VPN
  - e) VPN was subpoenaed by FBI, traced to internet cafe 500 ft from Ulbricht's house
  - f) July → border check from Canada found fake identification documents going to Ulbricht
3. Tor and Online Anonymity
  - a) Are there flaws in the Tor protocol?
    - Not much, the underlying technique is still somewhat safe
    - But, Tor does have problems:
      - Compromised exit nodes
      - Javascript vulnerabilities in browsers
      - "Bad apple attack"
  - b) Funding by the United States Government
    - 80% of funding was tied to the US government
    - Not direct, but funded by organizations funded by DoD
  - c) Has already been exploited
    - Child porn viewers:
    - FBI injected malware into a host which had Tor sites, found real IPs of child pornography viewers
4. Alternatives to Silk Road
  - a) Will these get shut down too?
    - Black Market Reloaded- a case study which says 'yes'.
    - Already taken down after the owner used a Virtual Private Server which was compromised
    - But, BMR is already back and doing business.
  - b) Atlantis- another Tor marketplace
    - Already shut down "due to security reasons outside of our control"
  - c) Sheep Marketplace
    - Still running and operational.
  - d) People always have to be involved, and people make mistakes.
    - But, with the ease of making a new site (even in Tor), sites may spring up faster than the FBI can shut them down.

Stack  
Overflow

Be careful  
Don't drift  
away from  
Silk Road.  
Focus on  
the #FAIL  
of  
Silk  
Road  
(and there  
were  
many)

- Delicate balance of Federal Government actions and illicit entrepreneurs.
5. Conclusions
- a) Silk Road was brought down for reasons not directly related to Tor.
    - So were all of the other marketplaces- Tor itself was not compromised.
  - b) At this point, it seems the Tor drug marketplace is a successful business model, as long as it is done with extreme care.
    - The FBI hasn't shut down any of these marketplaces because of Tor vulnerabilities
    - Human error, personal information, physical arrests all led to the shutdowns.
  - c) Is it safe to buy drugs online?
    - No. Sellers have been compromised (see reddit post), and if a server is subpoenaed, personal info (name, shipping address) could be accessed.
    - Encryption, encryption, encryption.
  - d) The battle of online anonymity is in full swing.

Sources(not yet complete):

[http://www.slate.com/blogs/crime/2013/10/17/black\\_market\\_reloaded\\_silk\\_road\\_s\\_biggest\\_competitor\\_shuts\\_down\\_after\\_site.html](http://www.slate.com/blogs/crime/2013/10/17/black_market_reloaded_silk_road_s_biggest_competitor_shuts_down_after_site.html)

<http://www.nbcnews.com/technology/how-anonymous-tor-users-compromised-child-porn-takedown-6C10848680>

<http://rt.com/usa/fbi-exploiting-tor-child-porn-842/>

<http://www.cnn.com/2013/10/04/world/americas/silk-road-ross-ulbricht/>

<http://www.bbc.co.uk/news/technology-24371894>

[http://www.reddit.com/r/SilkRoad/comments/1o6whp/attention\\_i\\_just\\_made\\_bail\\_as\\_a\\_buyer/](http://www.reddit.com/r/SilkRoad/comments/1o6whp/attention_i_just_made_bail_as_a_buyer/)

<http://www.theverge.com/2013/10/4/4802512/nsa-failed-to-compromise-tor-network-but-exploited-browser-vulnerabilities>

<http://www.ibtimes.co.uk/articles/514988/20131018/black-market-reloaded-silk-road-alternative-online.htm>

<http://www.ibtimes.com/atlas-illegal-online-drug-marketplace-forced-shut-down-due-security-reasons-outside-our-control>

<http://www.businessinsider.com/silk-road-alternatives-2013-10>

<http://www.bbc.co.uk/news/world-us-canada-17738207>



## Android Rootkits - Outline

### I. Introduction to the Android stack and rootkits

#### A. Android

1. Brief overview of Android components and mobile landscape (mobile market share, malware cases, etc.)

#### B. Rootkits

1. What is a rootkit?
2. Types of rootkits

- i. User mode, kernel mode, firmware, hypervisor (from highest to lowest ring location)

#### C. Rootkits' role in Android and why we should care

1. Android built on Linux kernel
2. Linux rootkits well documented, but much documentation is not up to date, and if it is, or if it is any good, it is probably in another language.
3. Linux rootkits within Android stack not well-documented.
4. Rootkits can be devastating.

### II. Linux rootkits primer

Also describe the  
Android  
architecture

## A. Hooking

### 1. Redirection of system calls (Malicious)

## B. Sys\_call\_table

## C. Kernel Symbol table

## III. Attack vectors

### A. Basic protections against modifying sys\_call\_table

#### 1. Read-only

#### 2. Don't export

### B. So how are we supposed to tamper with system calls!?

## IV. Attacking through Exception Vector Table

A. Execution branches to exception vector table when an interrupt or exception is received by the CPU (ARM - used by most phones (significantly lower power consumption))

### B. Getting sys\_call\_table address from software interrupt handler

### C. Determining size of sys\_call\_table

### D. Copying sys\_call\_table through /dev/kmem

### E. Modifying the software interrupt handler for hooking

## V. Intercepting system calls

### A. Which system calls should we modify?

B. Using dmesg to find system calls used by Android functionality

VI. What can we do now?

- A. View text messages
- B. Make phone calls
- C. Pretty much anything!

VI. Conclusion

- A. Methods to combat Android rootkits
  - 1. Digital signatures on all kernel code
- B. Call for increased security (awareness as well) within mobile computing industry (and in general)
- C. Thanks

Supporting material: Original rootkit code (assuming I can hack it :D)

Sources used thus far:

- <https://www.defcon.org/images/defcon-18/dc-18-presentations/Trustwave-Spiderlabs/DEFCON-18-Trustwave-Spiderlabs-Android-Rootkit.pdf>
- <http://phrack.org/issues.html?issue=68&id=6#article>
- <http://phrack.org/issues.html?issue=67&id=6#article>
- [http://w3.cs.jmu.edu/kirkpans/550-f12/papers/linux\\_rootkit.pdf](http://w3.cs.jmu.edu/kirkpans/550-f12/papers/linux_rootkit.pdf)
- <http://web.archive.org/web/20070610083142/http://rig.informatik.tu-chemnitz.de/docs/da-sa-tx/sa-dienelt.pdf>
- [https://media.blackhat.com/bh-ad-11/Oi/bh-ad-11-Oi-Android\\_Rootkit-WP.pdf](https://media.blackhat.com/bh-ad-11/Oi/bh-ad-11-Oi-Android_Rootkit-WP.pdf)

← what you can do: get an Android rootkit and analyze the source

Samuel Daniel  
10/31/2013  
Outline of Final Paper  
COMP 116

## 1. Introduction

- A. Why is cyber warfare important
  - i. How does it compare to nuclear war
    - a. Potential targets are similar
  - ii. Who is involved in cyber war
    - a. State vs. non state actors
- B. History of cyber warfare
  - i. Stuxnet
  - ii. Estonia
  - iii. Syrian Electronic Army
  - iv. Anonymous, etc.
  - v. Drones
  - vi. Criminal activity

## 2. To The Community

- A. Different and changing meanings of the term "hacker"
  - i. Technical community vs. non-technical community
- B. What we know and what we do not know
  - i. Governments are highly secretive about cyber activities
    - a. Snowden
  - ii. The importance of the open source community
- C. Cyberwar's relationship with civil liberties

## 3. Current State of The World

- A. Current Policy
  - i. U.S. policy
  - ii. E.U. policy
  - iii. China and Russia
- B. Technology
  - i. Stuxnet - open source
  - ii. Viruses loaded onto flash drives - U.S. CENTCOM
  - iii. DDoS
  - iv. Attacking Internet-connected infrastructure and non-computers
- C. Law
  - i. Issues of Attribution
  - ii. Lack of solid legal framework worldwide

## 4. Looking to the future

- A. The structure of the Internet
  - i. operating in an untrustworthy world
- B. New legal frameworks

Way too  
large of  
a scope.  
Can you  
focus  
on  
just  
the  
E.U.?

C. Cyber Terrorism vs. Cyber War

i. State vs. non-state actors

a. Hacktivists, extremists, etc.

5. Conclusion

SOURCES:

<http://jnslp.com/wp-content/uploads/2013/04/The-Dark-Future-of-International-Cybersecurity-Regulation.pdf>

<http://www.isn.ethz.ch/Digital-Library/Publications/Detail/?lng=en&id=153811>

[http://www.internetevolution.com/author.asp?section\\_id=515&doc\\_id=136047](http://www.internetevolution.com/author.asp?section_id=515&doc_id=136047)

<http://www.lawfareblog.com/2012/02/the-internet-kill-switch-debate/>

<http://www.cfr.org/internet-policy/internet-governance-age-cyber-insecurity/p22832>

[http://dspace.mit.edu/bitstream/handle/1721.1/1519/TPRC\\_Clark\\_Blumenthal.pdf](http://dspace.mit.edu/bitstream/handle/1721.1/1519/TPRC_Clark_Blumenthal.pdf)

[http://www.washingtonpost.com/world/report-ties-100-plus-cyber-attacks-on-us-computers-to-chinese-military/2013/02/19/2700228e-7a6a-11e2-9a75-dab0201670da\\_story.html?hpid=z1](http://www.washingtonpost.com/world/report-ties-100-plus-cyber-attacks-on-us-computers-to-chinese-military/2013/02/19/2700228e-7a6a-11e2-9a75-dab0201670da_story.html?hpid=z1)

<http://cyber.law.harvard.edu/events/luncheon/2013/05/edgar>

Supplemental Material:

Some sort of FYI website documenting past cyber attacks by location, type, and target

# AN ANALYSIS OF MOBILE MALWARE DETECTION TECHNIQUES OVERVIEW

**Submitted by: Aswathy Dinesh (adines01)**

Talk about Mobile malware and Malware Detection techniques and its strength and weakness.

## **Cover:**

- What is Mobile Malwares and type of malware?
  - A small description about mobile Malware.
  - Kind of malwares and their symptoms.
  - Who are the attackers (malware creator)?
  - What do attackers (malware creator) want?
- Malware detectors and how do they work?
- High-level overview of malware detectors and how it works on mobile devices?
  - What are the inputs they take?
    - Malicious behavior
    - Take program as an input and decide whether the program is malicious.
- Malware Detection techniques
- Review and analyze the malware detection techniques and also discuss its strengths and weakness
  - Static Analysis
    - Analysis of code without executing program.
    - Types: System call based, Static taint analysis, Source code analysis.
  - Dynamic Analysis
    - Monitoring the behavior of mobile application
  - Application Permission Analysis
    - Asks the user to grant or deny permission for the application based on the activities the application performs
  - Cloud based detection
    - Perform analysis on the cloud
  - Battery life Monitoring:
    - Observe energy consumption and detect malware.

Paper Outline: Google Glass and Wearable Technology

I. Introduction

- a) Explanation of a Google Glass
  - What it is (high level)
  - What hardware it contains
  - How it works (again, high level)
- b) Where the paper is going
  - Social Engineering
  - Coding for the Glass
  - Hardware and Hacking

What will  
your  
demo  
be?

II. Social Engineering Implications

- a) Wearable Technology
  - New idea, VERY insecure
  - Discuss risks inherent in having private information stored essentially in your glasses
  - No password locking as of yet
  - Technology is new and unprecedented, so all security exploits cannot possibly be accounted for yet
- b) Privacy
  - Through custom voice commands and other input, easy to take video and pictures
  - Essentially makes it *even easier* to take information from unsuspecting people

I'm  
curious

III. Coding and Custom Apps

- a) How to code for the Google Glass
  - explain Mirror API, general code process from computer / server >> Glass
  - What precautions Google has taken (no JS)
- b) POTENTIAL SUPPORTING MATERIAL:
  - App written for a server that will allow user to do something malicious
  - Actual, usable code that is compatible with a Glass
  - e.g. Take a name, or picture, and search for them on social media (stalking)
- c) Why is this a concern?
  - People can create malicious apps for personal use without being obvious
  - People can create malicious apps for other's use (this is a common problem to many app platforms, however)

really?

IV. Hardware and Hacking

- a) The device
  - Always on, no password, etc

## The Sybil Attack: Peer-to-Peer Network Vulnerability

- What Are Peer-to-Peer Networks?
  - No C&C server.
  - Distributed Hash Table
    - XOR distance metric
    - Circle metric (e.g. Chord)
  - Real World Application
    - BitTorrent
    - Botnets
    - Freenet
    - Web caching
  - Comparison to C&C architecture
    - Advantages/Disadvantages
- Kademlia
  - DHT protocol
  - Came up with XOR Metric
  - Modified version used by BitTorrent
  - Describe technical properties
- Sybil Attack
  - What is it?
  - What can it be used for?
    - BitTorrent example
    - Compromises data within network
  - How to do it
    - Code up simulation of a Sybil attack.
  - How to stop it
    - Trust-based reputation system
      - Best approach
      - Tonika - next-gen of Kademlia (stalled)
      - How to validate nodes?
      - Trade-off between security and openness.
    - Other approaches
      - TBD

So this  
is  
still  
huge today?

good



# COMP 116 Project Outline

Fall 2013

Albert Jose de Vera

ajdevera@cs.tufts.edu

Albert.de\_Vera@tufts.edu

1. Introduction
2. To The Community
3. Threats and Attacks
  1. Current Threats and Nature of Attacks
  2. Defenses Against Attacks
4. Web Application Firewalls
  1. Evolution of Web Application Firewalls
  2. Web Application Firewalls Today
5. Configuring and Using Web Application Firewalls
  1. Web Server and Web Application Setup
    1. Apache httpd (on Linux)
    2. Microsoft IIS (on Windows Server)
  2. Web Application Firewall Configuration and Usage
  3. Testing Exploits Against Web Application Firewalls
    1. Aqtronix Webknight (IIS)
    2. ModSecurity (IIS and Apache)
    3. IronBee (in development: development code available for testing)
    4. Commercial products (if available for testing)
6. Effectiveness of Web Application Firewalls – results from tests against the web application firewalls
7. Summary: Securing a Web Application – recommendations and sample configurations

People can read instructions on how to do this

## References:

M. Muthuprasanna, K. Wei, et. al. Eliminating SQL Injection Attacks – A Transparent Defense Mechanism. Proceedings of The Eight IEEE International Symposium on Web Site Evolution (WSE '06). 2006. ISBN 0-7695-2696-9.

R. B. Brinhosa, C. B. Westphall, C. M. Westphall. A Security Framework for Input Validation. Proceedings of The Second International Conference on Emerging Security Information, Systems and Technologies. ISBN 978-0-7695-3329-2 (Electronic).

D. Tsai, A. Y. Chang, et. al. Optimum Tuning of Defense Settings for Common Attacks on the Web Applications. Proceedings of the 43rd Annual 2009 International Carnahan Conference on Security Technology, ICCST 2009. ISBN 978-1-4244-4170-9.

P. Bisht, P. Madhusudan and V. N. Venkatakrishnan. CANDID: Dynamic Candidate Evaluations for Automatic Prevention of SQL Injection Attacks. ACM Transactions of Information and System Security. Vol 13. No. 2. Article 14. February 2010.

B. Sullivan and V. Liu. Web Application Security: A Beginner's Guide. McGraw-Hill. 2011. ISBN 978-0-07-177612-7 (Web).

C. Serrão, V. Díaz and F. Cerullo. Web Application Security. Iberic Web Application Security Conference, IBWAS 2009, Madrid, Spain, December 10-11, 2009. Revised Selected Papers. Springer: 2009. ISBN 978-3-642-16120-9 (Online).

## URLs:

J. Rafail. Cross-Site Scripting Vulnerabilities. CERT Document. 2001.  
[http://www.cert.org/archive/pdf/cross\\_site\\_scripting.pdf](http://www.cert.org/archive/pdf/cross_site_scripting.pdf)

<http://www.cert.org/advisories/CA-1995-04.html> NCSA HTTP Daemon for UNIX Vulnerability

Perhaps have one quick on the CTF VM.