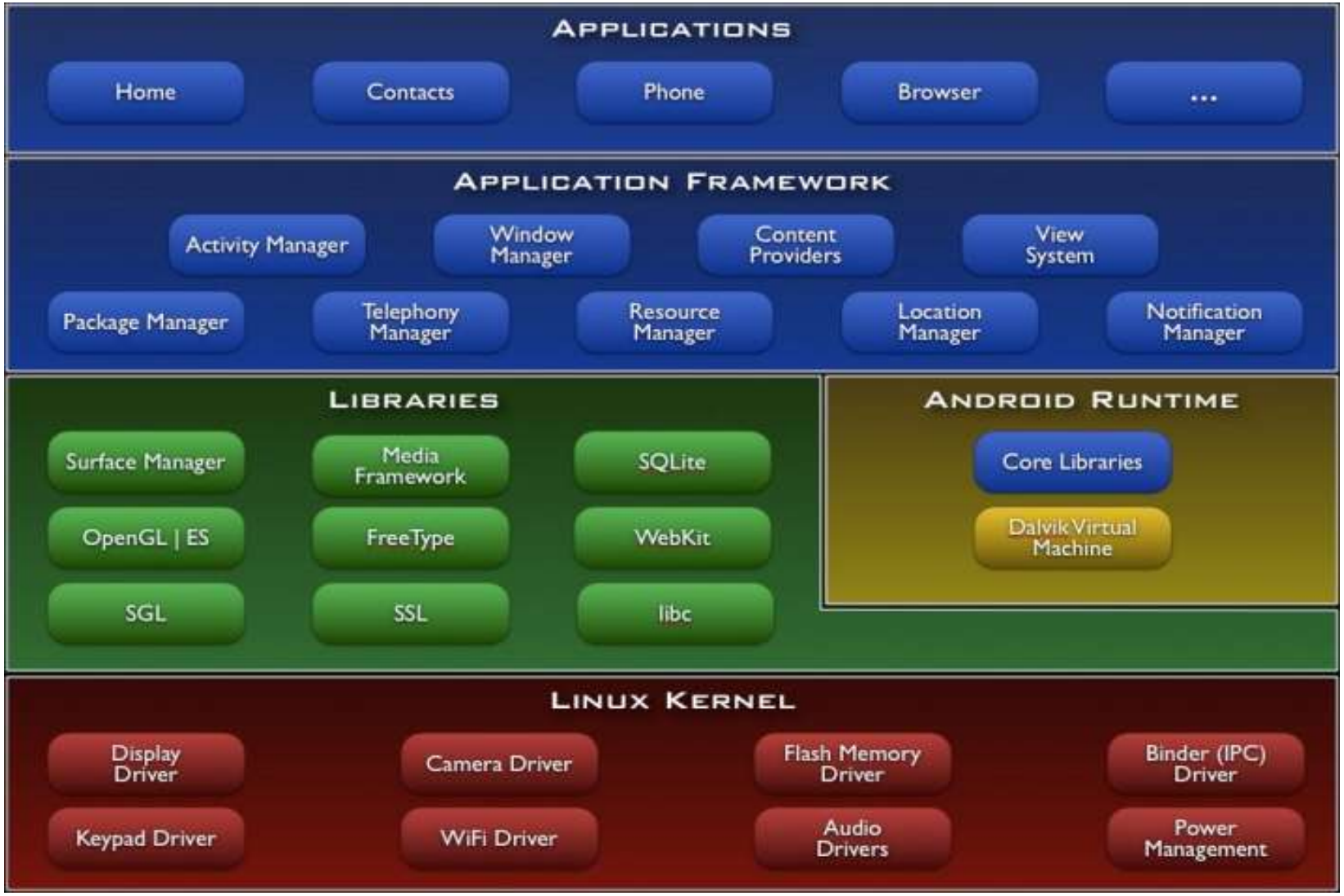
Android

By Collin Donaldson

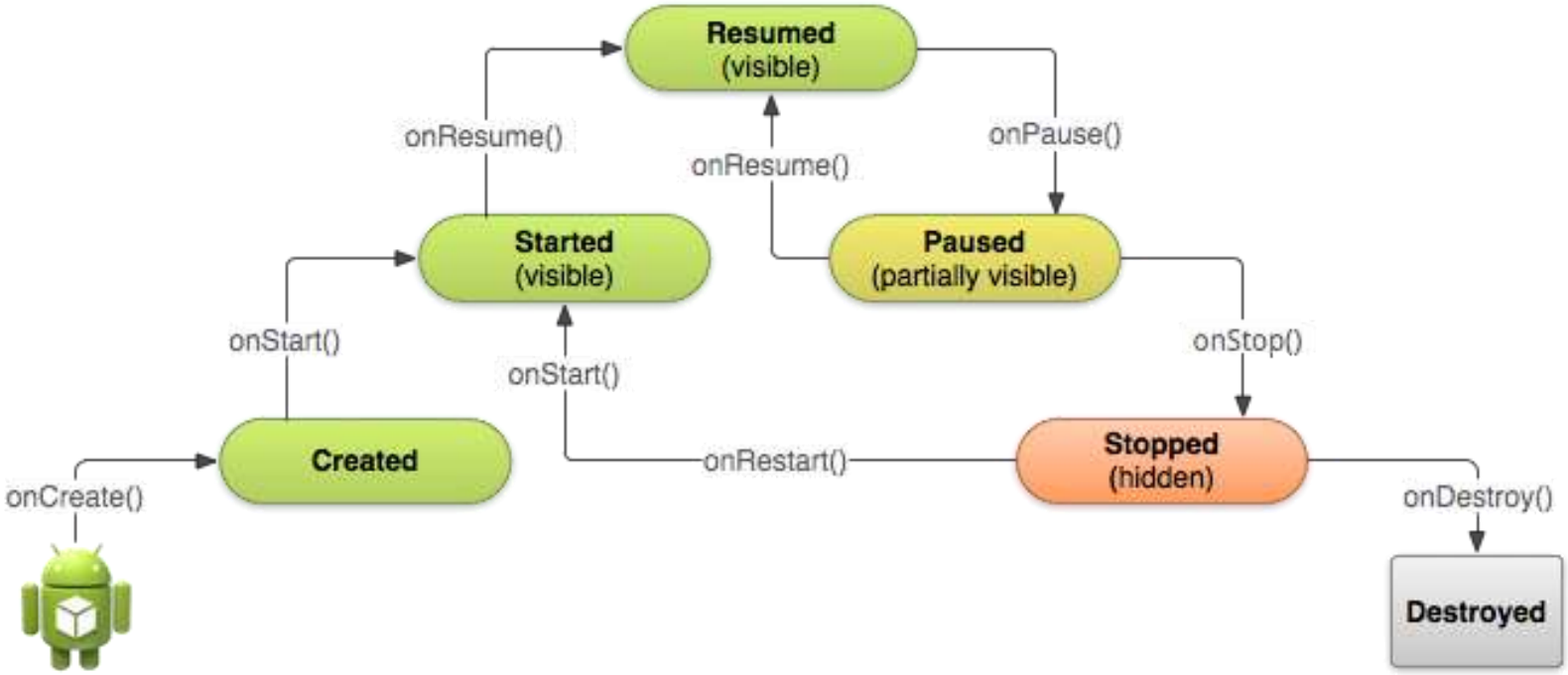
With Strong Contributions From:

Anthony Kopczyk

Architecture



Activity Life Cycle



Activity Life Cycle - onCreate

- Set the Activity's content View; Event Listeners
- Find references to any needed Views
- Passed-in Bundle allows a programmer to restore the Activity to its previous status

Activity Life Cycle - onStart

- Called when the Activity becomes visible
- Initialize any properties requiring information from the Window and contained Views

Activity Life Cycle - onResume

- Called when the Activity is visible and in the foreground
- Initialize ability for user to interact with the Activity

```
@Override
public void onResume() {
    super.onResume();  // Always call the superclass method first

    // Get the Camera instance as the activity achieves full user focus
    if (mCamera == null) {
        initializeCamera(); // Local method to handle camera init
    }
}
```

Activity Life Cycle - onPause

- Called when the Activity is no longer the foreground
- Release system resources

```
@Override
public void onPause() {
    super.onPause();  // Always call the superclass method first

    // Release the Camera because we don't need it when paused
    // and other activities might need to use it.
    if (mCamera != null) {
        mCamera.release()
        mCamera = null;
    }
}
```

Activity Life Cycle - onStop

- Called when Activity is no longer visible
- Perform larger operations like writing to a database
- Save Activity's state for onStart

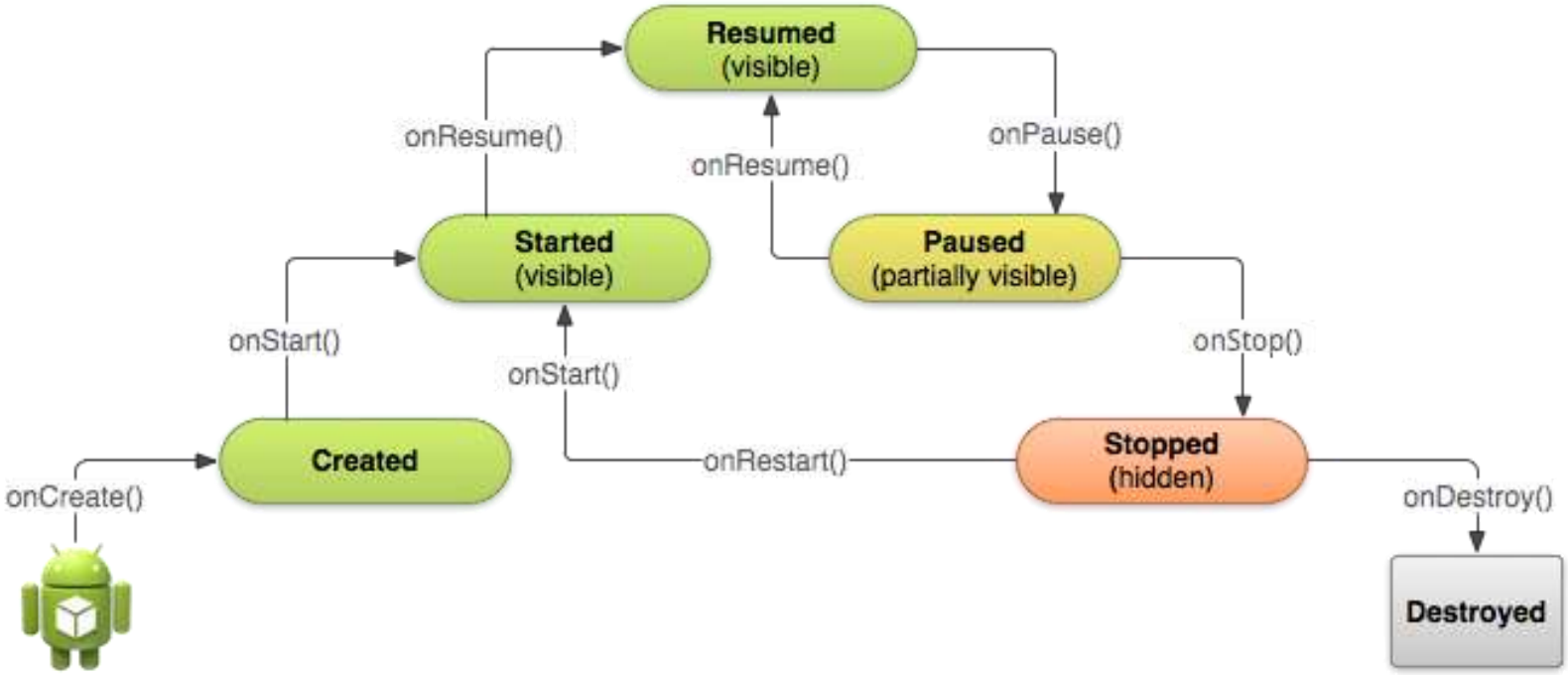
```
@Override
protected void onStop() {
    super.onStop();  // Always call the superclass method first

    // Save the note's current draft, because the activity is stopping
    // and we want to be sure the current note progress isn't lost.
    ContentValues values = new ContentValues();
    values.put(NotePad.Notes.COLUMN_NAME_NOTE, getCurrentNoteText());
    values.put(NotePad.Notes.COLUMN_NAME_TITLE, getCurrentNoteTitle());

    getContentResolver().update(
            mUri,    // The URI for the note to update.
            values,  // The map of column names and new values to apply to them.
            null,    // No SELECT criteria are used.
            null     // No WHERE columns are used.
            );
}
```

Activity Life Cycle - onDestroy

- Called when the system is in need of resources
- Last chance to free resources and avoid memory leaks

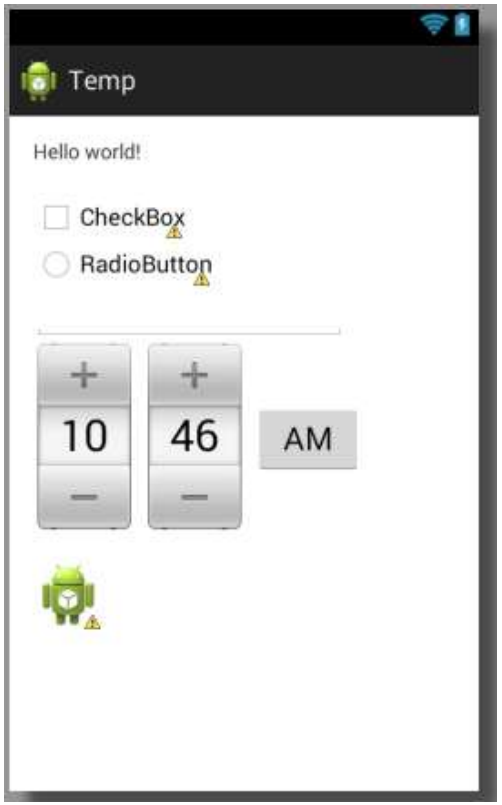
```
@Override
public void onDestroy() {
    super.onDestroy();  // Always call the superclass

    // Stop method tracing that the activity started during onCreate()
    android.os.Debug.stopMethodTracing();
}
```

Activity Life Cycle



Views

- A building block for UI components
- Responsible for drawing and event handling
- Each View has an id
- findViewById(int)



Text box, check box, radio button, time picker, and image view

XML

- Eclipse IDE
- Uses XML files to set up the mobile application

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.temp.MainActivity$PlaceholderFragment" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

</RelativeLayout>
```

Android Layout File

XML - Android Manifest

- Contains properties of the application
- Permissions, SDK, Icon, Activities

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.kopczyk.hos"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-sdk
        android:minSdkVersion="11"
        android:targetSdkVersion="18" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/hos_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme"
        android:hardwareAccelerated="true"  >

        <activity
            android:name="com.kopczyk.hos.ActivityHos"
            android:label="@string/app_name"
            android:screenOrientation="landscape" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Android Manifest File

XML - Layout

- Defines the layout of the Activity
- Set View id values
- Could achieve the same results through java code

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.temp.MainActivity$PlaceholderFragment" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

</RelativeLayout>
```

Android Layout File

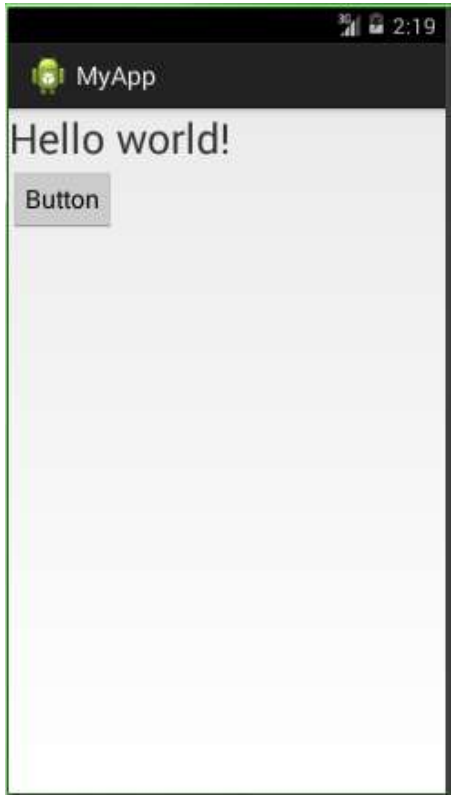
XML – Other Files

- Menu – Defines the Menu to bring up
- Dimens – Defines dimensions with names and values
- Strings – Defines strings with names and values
- Lint – Defines exclusion or customization of lint checks
- Styles – Defines the style to use in the Activity
- Attrs – Defines custom attributes that may be used in XML Layout files

Input Events

- OnClickListener
- OnLongClickListener
- OnFocusChangeListener
- OnKeyListener
- OnTouchListener
- OnCreateContextMenuListener

Input Events



Pre-Click → Post-Click

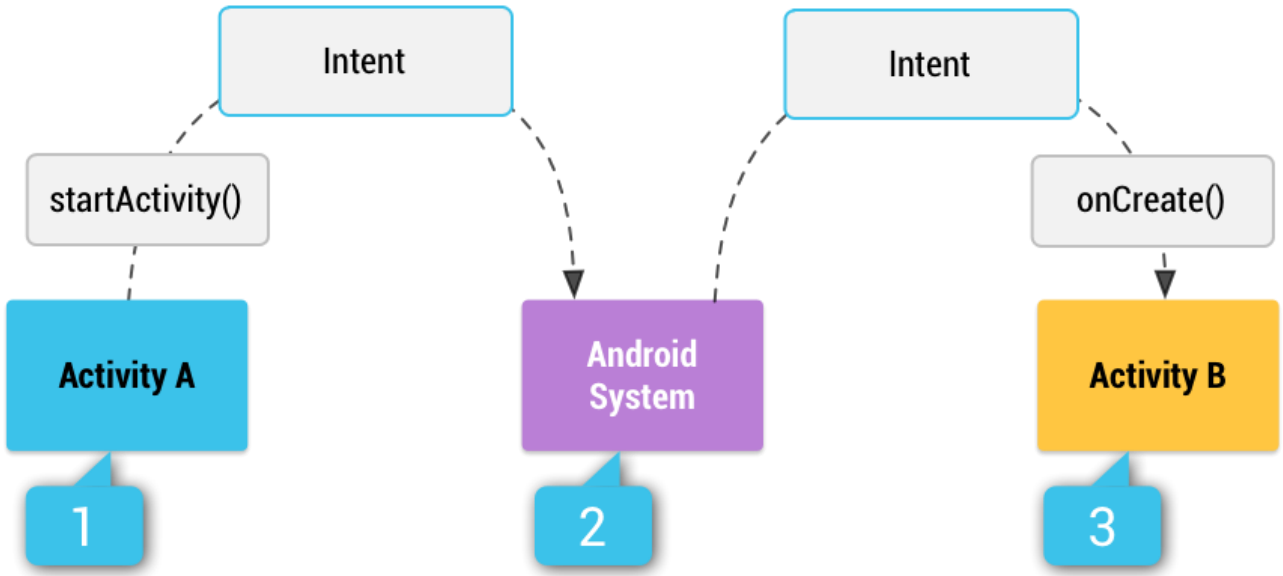
```
package com.example.myapp;

import android.app.Activity;

public class MyActivity extends Activity {

    Button button;
    TextView textView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Set the Content View
        setContentView(R.layout.activity_my);

        // Find Views
        button = (Button) (findViewById(R.id.button1));
        textView = (TextView) (findViewById(R.id.textbox1));

        // Set the OnClickListener
        button.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View arg0) {
                // Change TextView's color
                textView.setTextColor(Color.GREEN);
            }
        });

    }
}
```

Intents

- Starts an activity or service
- Service – operates in the background without a UI
- Intents can be sent to other Apps
- sendBroadcast()
- sendOrderedBroadcast()
- sendStickyBroadcast()

Intents – Explicit vs. Implicit

- Explicit Intents specify a component to start.
- Implicit Intents give a general action to perform.



Intents - Intent-Filter

- Specified in the Manifest file
- Contains the types of Intents the app wants to receive
- Allows one app to send an Intent to another
- If no intent filters are specified the activity may only be started with an explicit Intent
- To ensure security, always use explicit intents when starting a Service
- Users can not see when a Service starts

Permission

- Allows developers to use security features
- Provides additional capabilities to consumers that otherwise would be impossible

"A central design point of the Android security architecture is that no application, by default, has permission to perform any operations that would adversely impact other applications, the operating system, or the user"

Permission

- When an Application is installed the consumer must accept the permissions requested by an application
- Permissions are defined in the Manifest file

Permission

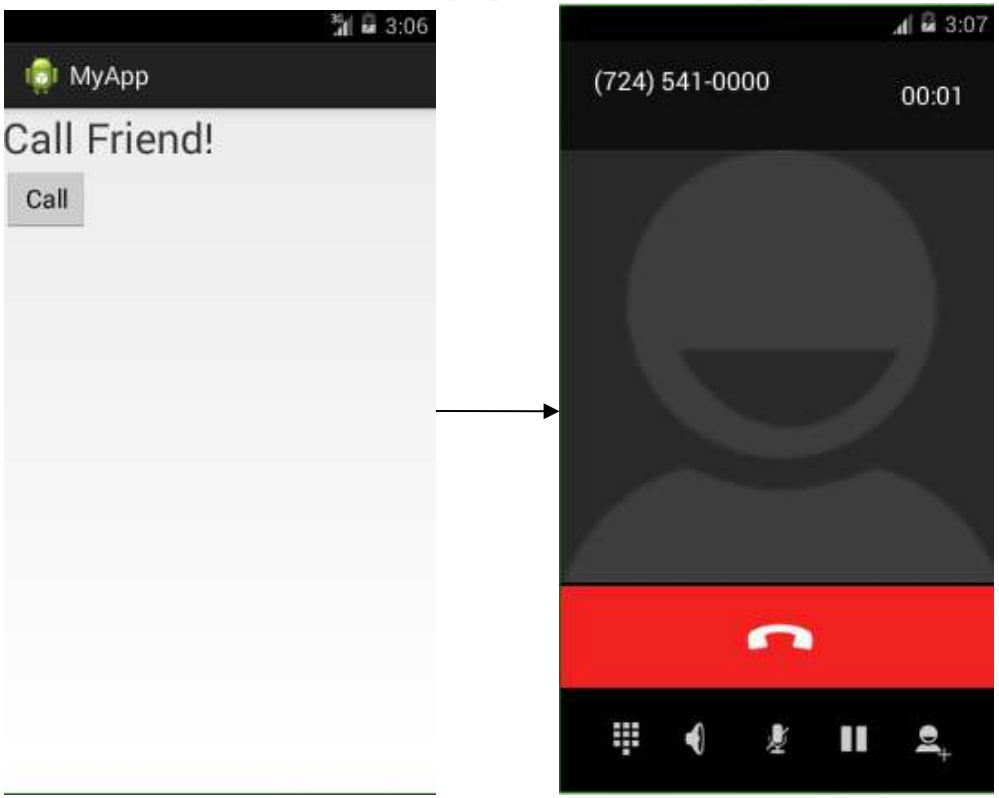
- Facebook Messenger
  - Identity
  - Contacts/Calendar
  - Location
  - SMS
  - Phone
  - Photos/Media/Files
  - Camera/Microphone
  - Wi-Fi Connection Information
  - Device ID & Call Information
- Angry Birds
  - •Identity
  - •In-app purchases
  - •Location
  - •Photos/Media/Files
  - •Wi-Fi Connection Information
  - •Device ID & Call Information

Permission

- Camera/Microphone
  - Allows consumers to use video chat
- Photos/Media/Files
  - Allows consumers to send pictures they have previously taken
- In order to give access, one must become more vulnerable – like opening ports on your router

Permission

- Using Intents and Permissions a developer can make his/her app call a phone number.



Permission

```
import android.app.Activity;

public class MyActivity extends Activity {

    Button button;
    TextView textView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Set the Content View
        setContentView(R.layout.activity_my);

        // Find Views
        button = (Button) (findViewById(R.id.button1));
        textView = (TextView) (findViewById(R.id.textbox1));

        // Set the OnClickListener
        button.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View arg0) {
                // Call friend
                Intent intent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:" + "7245410000"));
                startActivity(intent);
            }
        });

    }
}
```

Activity Class

Permission

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapp"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="14"
        android:targetSdkVersion="19" />

    <uses-permission android:name="android.permission.CALL_PHONE" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.myapp.MyActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Manifest File

Rooting/JailBreaking

- Rooting is the process of gaining root (a.k.a. administrator or super user) to a smartphone.
- By default, all smartphones only give user's "guest" privileges.
- This is for both safety reasons and to prevent users from doing things developers/carriers don't like (i.e. getting rid of their bloatware).

Pros and Cons to Root

Pros

- Download more apps and use existing apps to fullest potential
- Flash custom ROMs
- Access locked hardware/software features
- Tune performance
- No more bloatware
- Wi-Fi/Bluetooth Tethering
- Use apps designed for other phones/carriers
- Install apps to an SD card

Cons

- If done incorrectly, can possibly brick phone
- Voids any warranties you have (even if you reverse the root)
- Less stable/more bugs

General Security Vulnerabilities

- Flaws in Android OS itself
- Flaws in phone software/firmware
- Conventional browser based virus
- Vulnerabilities within downloaded apps
- Unconventional attacks (injecting code into accelerometers i.e.)

Specific Vulnerabilities

- Backdoor.AndroidOS.Obad.a does not have an interface and works in background mode, making it difficult to analyze, but that was only part of the challenge, according to Unuchek. The application exploits an error in the DEX2JAR software – generally used by researchers to convert APK files into the Java Archive (JAR) format) – that disrupts the conversion of Dalvik bytecode into Java bytecode and makes it difficult to run a statistical analysis of the Trojan.

- Obad.a also targets an error in Android's processing of the AndroidManifest.xml file, which exists in every Android application to describe the application's structure, define its launch parameters and more. Although Obad.a modifies AndroidManifest.xml so that it doesn't comply with Google standards, the vulnerability enables it to still be processed correctly, complicating any attempt to run dynamic analysis on the application.

Next Time

- We will use a Metasploit (with a specific module) to attack an android device.
- The “android device” will be a virtual android machine running on an emulator
- We may also write a virus and Python and deploy it to a device.

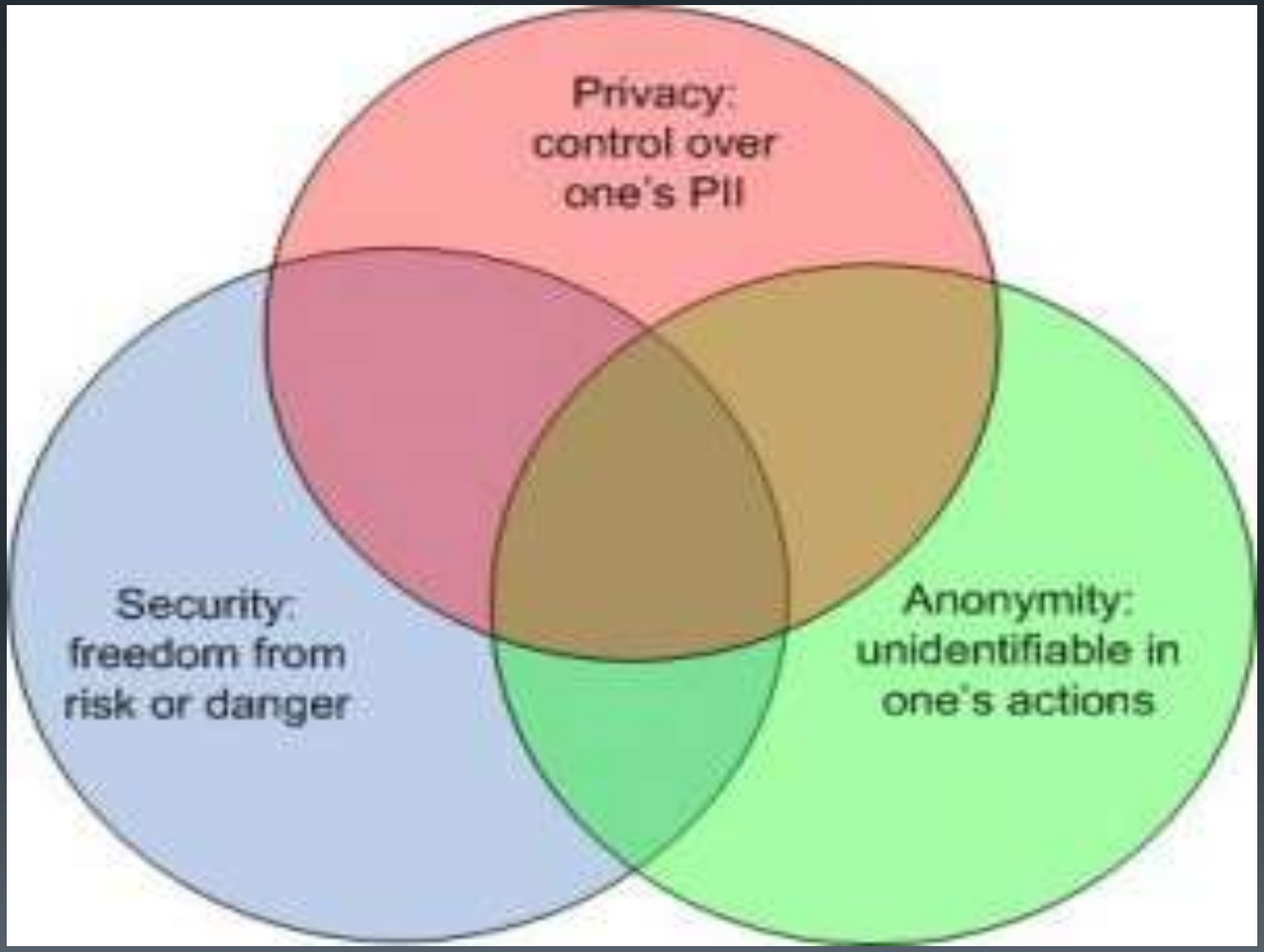
Anonymous Internet Browsing

by Collin Donaldson

Outline

- Overview
- Definitions and Explanations
- Part 1: How it all works
- Overview of Network Communication
- Overview of Network Traffic
- Overview of Internet Browsing
- Part 2: How to anonymize it
- Private Browsing and beyond
- DuckDuckGo
- Proxies (Onion Routing) and VPN
- Mobile Security
- Sources

Definitions: Ideals Not Realities!



Privacy vs Security vs Anonymity

- Privacy (P): “I control my information”.
- Anonymity (A): “A third party cannot associate my information with me.”
- Security (S): “My information is safe from interference (harm)”.
- Not mutually exclusive
- Can be combined (protection in depth)
- Single point of failure: the human

Hypothetical Example

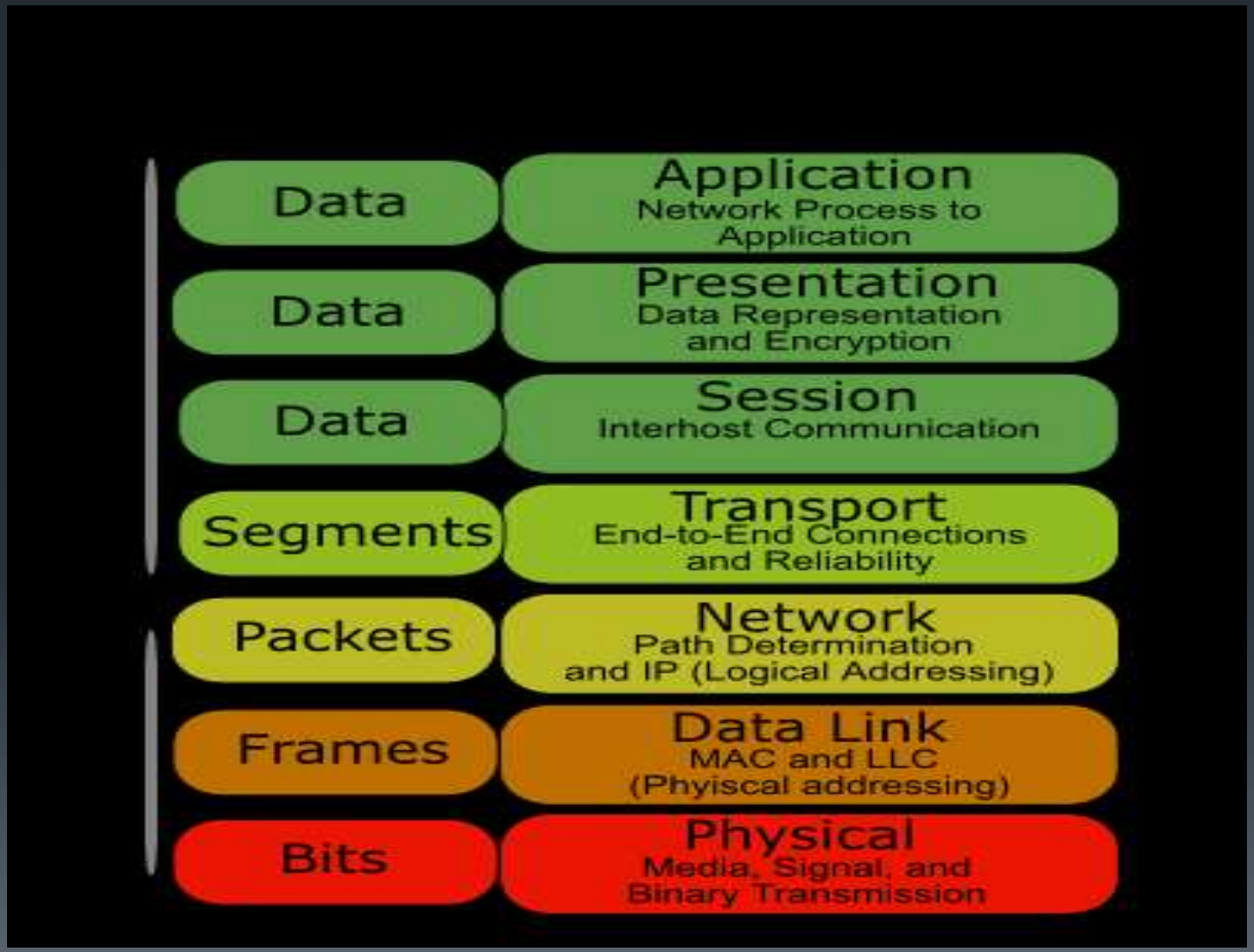
- A video file (VF) resides in your computer.
- Assume the computer is always offline, is immune to digital forensics, and can only be accessed by you.
  - VF: S, P, A
- You allow others to use your computer.
  - VF: !S, !P, !A
- You encrypt the file (homebrew)
  - VF: S, !P, !A
- You implement user control, others cannot access the file
  - VF: S, P, !A
- You hide the file from other users
  - VF: S, P, A

The Problem

- Networks greatly complicate security, privacy, and anonymity
- A known (not anonymous) file breaches privacy and risks security
- A non-private file breaches anonymity and risks security
- An insecure file risks privacy and anonymity
- Uploading that video file from the previous example to a website like YouTube throws anonymity right out the window

Crash Course in how Networks and Internet Browsers Work!

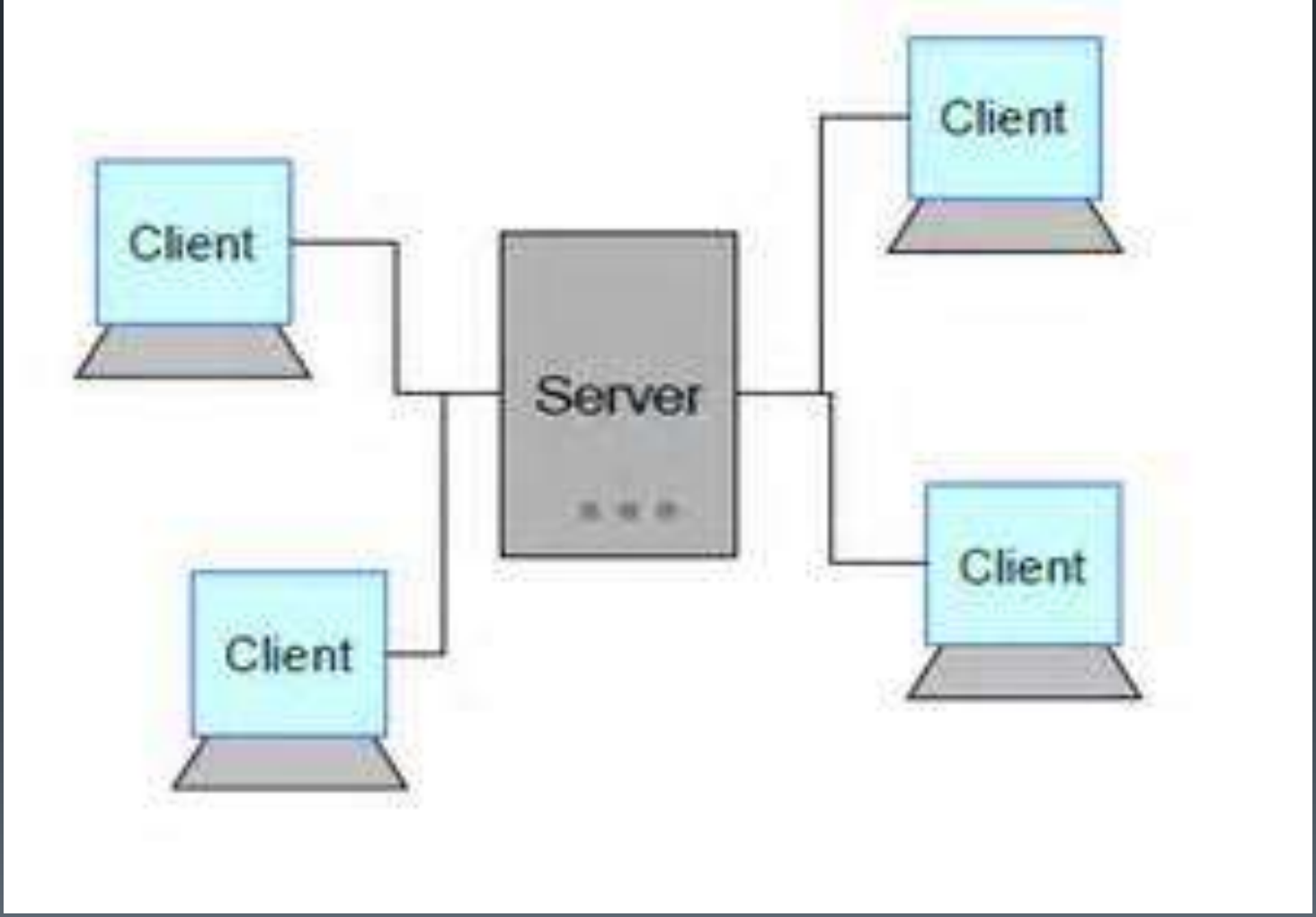
Network Communication: The Open Systems Interconnect (OSI) Model



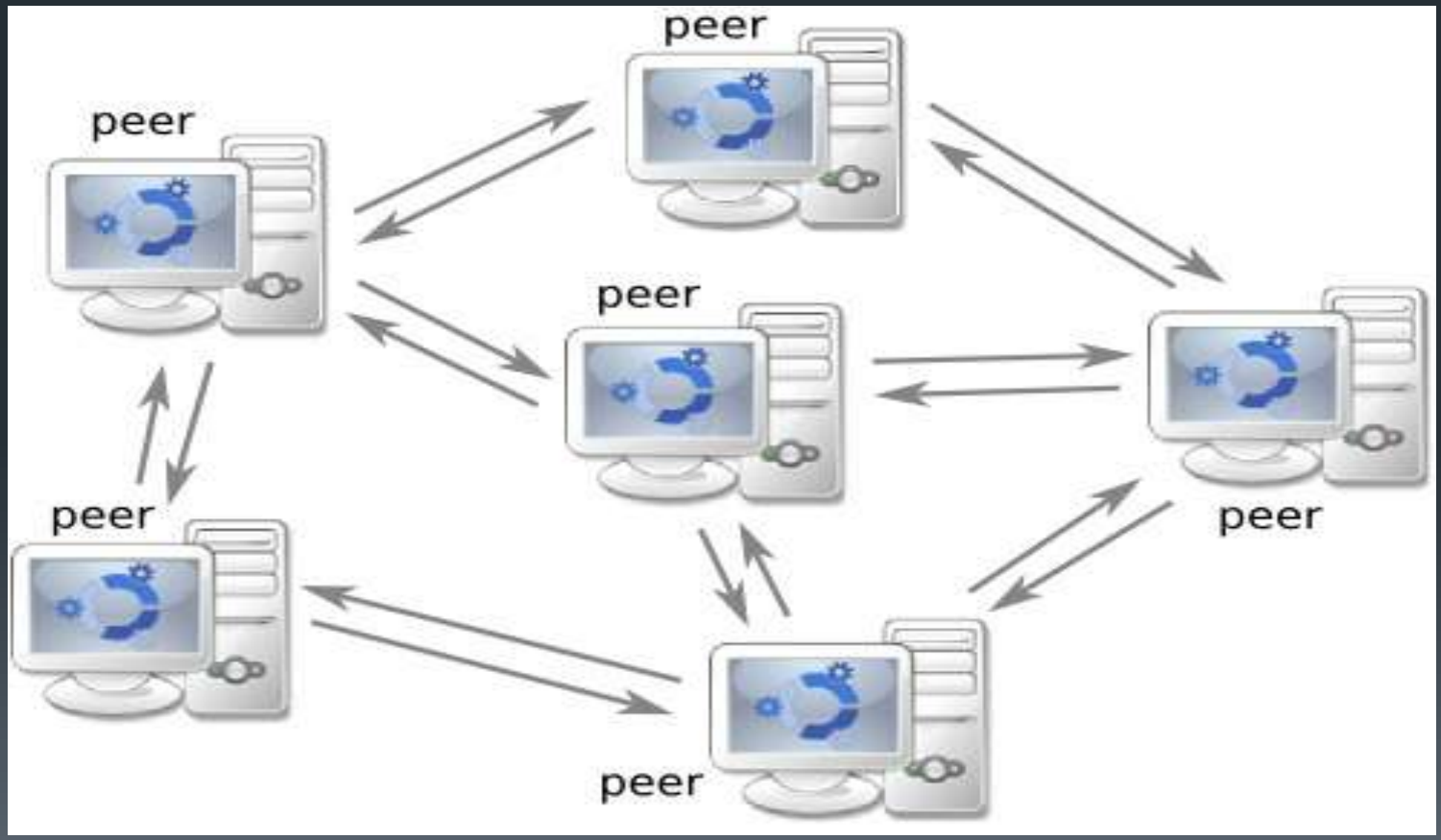
NOTE!!!!

- This presentation will focus on the Application, Presentation, Session, Transport, and Network Layers.
- The Data Link and Physical Layers are beyond the scope of this presentation, but are still important!
- For example: Companies like Verizon and AT&T have been caught using Unique Identifier Headers (UIDHs) which act like “permanent cookies” via all 7 layers.

Client-Server Model



Peer to Peer (P2P)



Network (IP) Traffic

Transport Control Protocol (TCP)

1. Ex: HTTP, FTP, SMTP
2. Connection oriented
3. Reliability > Speed
4. Heavyweight (requires 3-way handshake)
5. Usage: Small Data
6. Target Practice Analogy:

   Stop and Pop

User Datagram Protocol (UDP)

1. Ex: DNS, DHCP, VOIP
2. Connection-less
3. Speed > Reliability
4. Lightweight (no handshake)
5. Usage: Big Data
6. Target Practice Analogy:

   Spray and Pray

Web Browsers and the Internet

- 1. Your web browser is a client that allows you to access the world's largest server, the internet.
- 2. The internet uses a service called the World Wide Web to make it easy for clients (browsers) to view its content.
  - URLS, IP addresses, search engines (web crawlers), etc.
- 3. Web browsers display information by interpreting HTML references accessed via the internet. The HTML acts as pointers to the information you actually want to access.
- 4. The HTML's coding tells the browser how to interpret the information (i.e. file format)
- 5. Browsers need to be configured with plugins to view certain kinds of information.
- Ex: Adobe Flash, Active X, Java, JavaScript, etc.
- 6. Previous information is stored on your browser for future easy access (cookies, history, etc.).

Private Browsing

- Unlike normal browsing, no information is stored about you
- Private Sessions are “sandboxed” from normal sessions
- Plugins may or not be private
- Still vulnerable to:
  - Search engines
  - Websites that collect/share info about you (read: most websites)
  - Malware
  - Internet Service Provider (ISP) surveillance
  - Physical surveillance

Private Browsing++

- NoScript: Control exactly what scripts (plugins) run on your computer
- AdBlock Plus: Keep ads and adware from executing on your computer
- Abine's Blur: Anti-Trackers, Autogenerated Proxy Email Addresses, Passwords, and Credit Card Numbers,
- ShodanHQ and ScamAdvisor.com: Check where a website is and if it is legitimate.
- CCleaner and others: Customizable local data management

DuckDuckGo

https://www.youtube.com/watch?v=GJ-5A9xqxBY\

ZERO Data Collection

Less Spam

Faster results

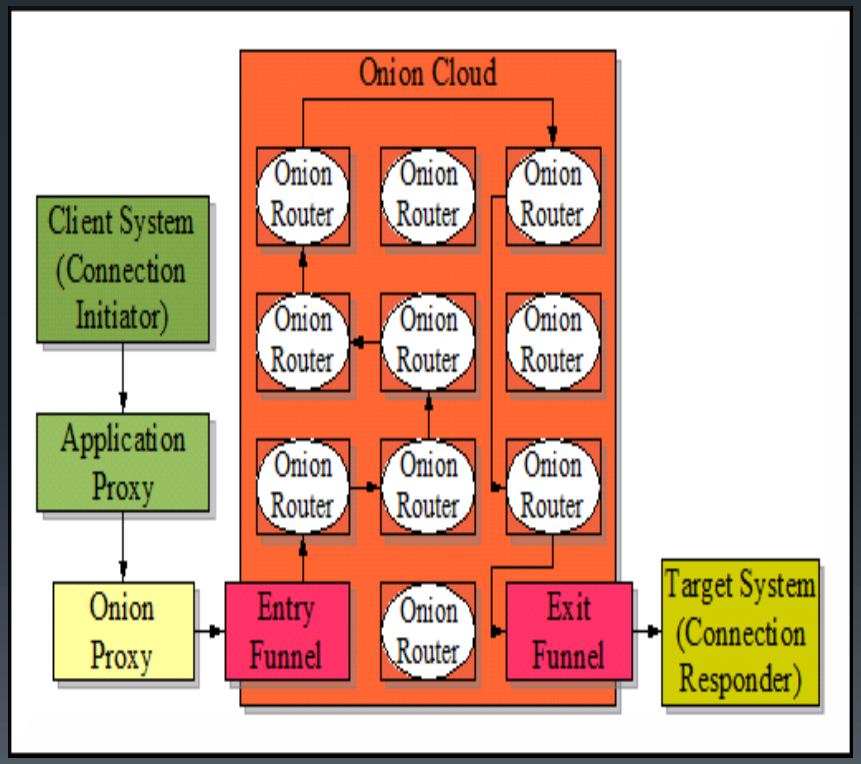
Uses Google's Monopoly Against It

Sponsors:
Mozilla Firefox
Apple Safari
Microsoft Bing
Yahoo Search
Tor Foundation

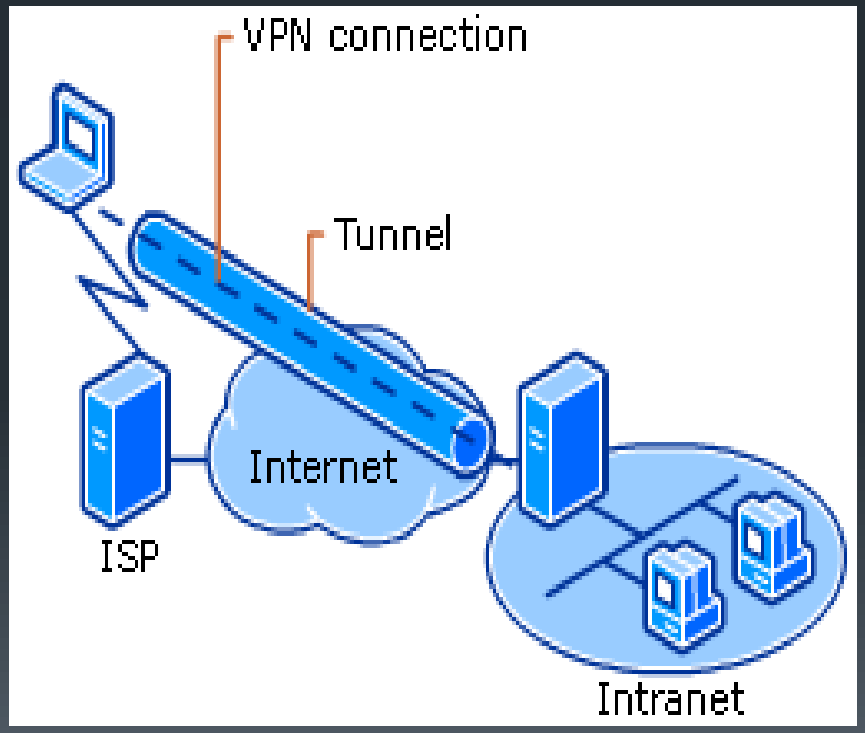
And Many More!

Proxies vs Virtual Private Networks

Proxies (Tor)



VPN



Onion Routing and VPN

Onion Routing

1. Free!
2. Main Architecture: P2P
3. Do It yourself
4. Priority: Privacy and Anonymity
5. Protocol: All TCP (effects speed)

VPN

1. $$$
2. Main Architecture: Client-Server
3. Generally All-In-One (commercial)
4. Priority: Security and Privacy
5. Protocol: TCP and UDP

Mobile Anonymity

- Anonymity does not automatically transfer to mobile devices!
- You have to manually configure everything again…..
- Browsers: DuckDuckGo (or configure others browsers)
- Proxy: Orbot
- VPN: OpenVPN
- Root/Jailbreak for full control!

Sources (URLS and titles)

- Computer Networks: A Top Down Approach (Pearson, everything, especially Ch.18 how web browsers work)
- https://www.torproject.org/about/overview.html.en (Tor)
- http://technet.microsoft.com/en-us/library/cc779919%28v=WS.10%29.aspx (VPNs)
- http://www.howtogeek.com/117776/htg-explains-how-private-browsing-works-and-why-it-doesnt-offer-complete-privacy/?PageSpeed=noscript (Private Browsing)
- http://support.microsoft.com/kb/103884 (OSI model)
- http://www.diffen.com/difference/TCP_vs_UDP (TCP/UDP)
- http://www.wired.com/2014/10/verizons-perma-cookie/ (UIDHs)
- https://duckduckgo.com/ (Research and DDG info)
- https://www.youtube.com/watch?v=GJ-5A9xqxBY (YouTube, DDG, Open Labs)

Sources (Images and titles)

- http://www.concurringopinions.com/wp-content/uploads/2011/01/privacy-security-anon1-300x253.jpg (Venn Diagram)
- http://4.bp.blogspot.com/-CUGfaG1zGzQ/Td4NpWHPKBI/AAAAAAAAAC8/G2TbB88zzDY/s1600/osi-model-7-layers.png (OSI)
- http://media-cache-ec0.pinimg.com/736x/ae/46/a5/ae46a5f95b9ce1d30f0b130aa4b5b257.jpg (NSA eyeball)
- http://www.caribbeanbusinesspr.com/fotos/cyber-soldier.jpg (Cyber soldier)
- http://img.youtube.com/vi/70lqb7v89Vg/0.jpg (Anonymous guy)
- http://www.game-en-co.nl/wp-content/uploads/2013/02/watch_dogs_wallpaper_by_neosayayin-d563o6l.jpg Watchdogs wallpaper
- http://www.crash-course.org.uk/wp-content/uploads/2012/12/Crash-Course-logo-new.jpg (Crash Course)
- http://www.chequeprinting.net/manual/server-client-setup.php (Client-Server)
- http://ten7023.patrickplante.org/archives/247 (DuckDuckGo)
- http://ntrg.cs.tcd.ie/undergrad/4ba2.05/group10/index.html (Onion Routing)
- https://guardianproject.info/wp-content/uploads/2010/03/orbot-on.jpg (Orbot)
- http://toniinfo.com/wp-content/uploads/2010/08/p2p.jpg (P2P)

On/Offline Countersurveillance

By Collin Donaldson

The Difficulty of Modern Countersurveillence

- Cyber-physical/Embedded System and Cloud Proliferation
- Big Data marketization (Data Brokers)
- The paradox of countersurveillance (Info in, Info out)
- Increased Intelligence Gathering and Sharing
  - Between Governments, Companies, etc.
  - Doxing (Anonymous, "Human Flesh Search Engine")

Countermeasures

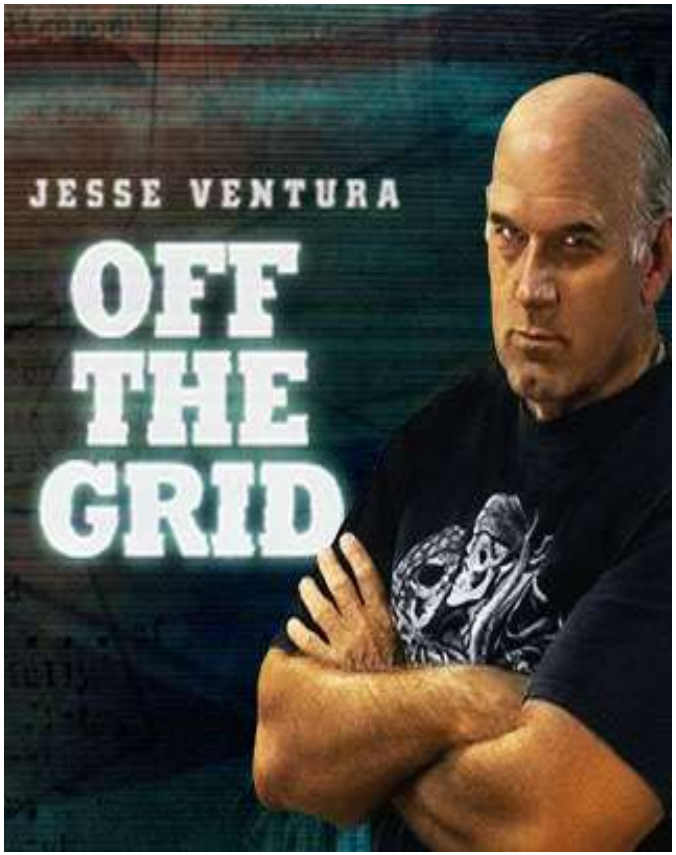
- First Step: Decide how private you want to be
- Second Step: Discover specifically who tracks you
- Third Step: Control your privacy

Step 1: Ask Yourself

- Do you care how visible your social media posts are?
- How about your email address?
- Location?
- Search history/preferences?
- Purchase history?
- Financial information?
- Decide approximately what you wish to safeguard

Step One: Ask Yourself (cont.)

- How much convenience will you sacrifice for PSA?
  - i.e. The cloud is more convenient than physical storage, but is also less PSA



Step 2: Find out who tracks you

- Services like Disconnect and Mozilla Lightbeam will overtime collect information on people that try to follow you

- Software like NoScript can block all scripts running on a page, but will also display all the trackers looking at

Step Three Part One: Stop Current Privacy Violations

- Change your social media preferences (can automate this with software like AVG's Privacy Fix)
- Use services like Abine's Blur and Dashlane to create and maintain proxy emails, passwords, debit cards, and more
- When you have to give information (i.e. security questions) falsify them, when legal
  - Consider creating different professional/personal personas

Step Three Part One: Control Your Current Privacy

- Check privacy policies, especially for freeware
- Default to
  - Private Browsing
  - Anon Browsing: DuckDuckGo, Startpage, Disconnect Search, Ixquick
  - HTTPS
  - Anti-tracking
  - Block 3rd Party Cookies

Step Three Part One: Control Your Current Privacy

- Route traffic through a VPN or Tor (may be covered more in depth in future meetings)
- HTTPS secure emails, or use alternative emails (proxies, homebrews, privacy-dedicated like Riseup or MyKolab)
- Use a self-destructing text/chat service
- Limit location tracking
- Turn off Wi-Fi and Bluetooth when not in use
- Use PC-level style security for your mobile devices

Step Three Part One: Control Your Current Privacy

- Pay with cash or disposable debit cards
- Make sure no one can see/hear your keystrokes when entering a PIN or password
- Keep devices and documents relatively hidden and secured
- Enable device tracking in the event of theft/loss
- Defense in Depth

Step Three Part Two: Control Already Exposed Information

- Delete unused "orphan" accounts
- Use software or shell commands to wipe old data
- Adapt privacy-friendly practices
- Delete records of credit cards at existing companies that don't need it
- Opt-Out of data broker knowledge list

Data Brokers to Opt-Outs

- NAI (Network Advertising Initiative)
- DMAchoice (online and off)
- DAA Ad Choices
- Spokeo
- Pipl
- ZoomInfo
- Whitepages
- PeopleSmart
- CheckPeople
- BeenVerified
- Intelius
- PublicRecords360
- ZabaSearch
- US Search
- PeopleFinders
- PeekYou

Alternatively: Automated Opt-Out Services (paid)

- Safe Shepard
- Reputation Defender
- Delete Me

Internships and Careers in IA

By Collin Donaldson

Types of Employers

- Corporations
- Mid-sized Companies
- Start-Up Companies
- Universities
- Government Agencies
- Think outside the box, a lot of different types of places need Computer Science majors! (example: weather stations need people to program their weather software and websites)

Potential Positions

- Software Engineer/Developer
- Web Developer
- Network Security Administrator
- Network Engineer/Architect
- Database Administrator
- Analysts (systems, data, network, etc.)
- Computer Forensic Scientist
- A specialization in Information Assurance is useful in all of the above positions

Courses to Remember

- COSC 110, 210, 220, 250, 300, 310, 319, and most 400 level courses for Programming/Theory COSC 316,
- COSC 345, 356, 362, and 454 for Networking/Security
- COSC 341 for Databases
- COSC 365 for Web development
- COSC 425 for Digital Forensics
- CRIM 321 and 323 Look Good too!
- Additional things to learn: new languages (especially different paradigms like dynamic and scripting), different software packages (i.e. Kali Linux), rising trends and useful legacy software, etc.

Resume

- Build your resume by developing new skills, taking more classes (at IUP and online at places like Coursera), and gaining real experience
- Make sure your resume grabs the person's attention
- List things the employer is looking for and will likely hire you for on top
- Include your accomplishments throughout, just do not get carried away
- Spice up mundane sounding positions/feats without stretching the truth
- Carefully proofread (no typos, awkward phrasing, etc.)
- The truth can be stretched a little, but do not outright lie!

Sample Resume and Cover Letter

- http://www.bestsampleresume.com/sample-student-resume/computer-science-student-resume.html
- http://www.resumebucket.com/sample-cover-letters/Computer-Science/Sample-Computer-Science-Cover-Letter.html

Interviews

- Get there early to ensure you are not late
- Dress the part
- Be polite and amiable
- Be prepared for both technical and open response questions
- Presenting a portfolio is a good idea, that way the employer could measure your abilities cumulatively on your terms opposed to on the spot.
- Radiate a desire to learn, work with others, and better the company.

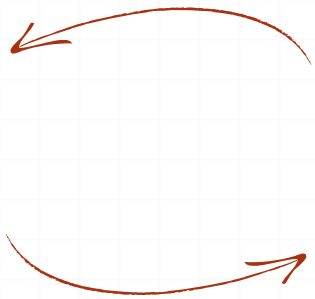
Tips

- [Social] Networking is important, do it smartly
- Dig around, look at names big and small
- Be a jack of all trades but master of one
- Never stop learning, the field is always moving
- Research the organization before going for an interview
- Experience is key to success
- Show your thought process!
- REMEMBER: Job postings are idealistic, not realistic

Job Requirements Translation

What it says

1. Excellent communication skills
2. 0-2 years experience
3. Requires experience in: IBM Rational Products, Microsoft SQL Server 2008, Visio, Office, .NET framework, Machine Learning/AI, Cisco 7000 series routers, etc.



What it means

1. Be able to communicate like a normal human
2. Entry level, experience preferred
3. Will probably hire if you have the top requirement, anything more is a big plus

Tips (continued)

- Maintain a Circum Vitae if you want to go into academia or research-related positions
- Portfolios > Resume/Cover Letter
- Prepare for technical questions/live demo of skills
- Have goto responses to common questions and even oddball ones
- In an interview, try to identify who is who
- Also, keep the employers talking if you can (this is a conversation, not a monologue!)
- Nepotism can be a friend or an enemy

Acronyms

- SDLC = Software Development Life Cycle
- COTS = Commercial Off The Shelve System
- DBM = Data Base Manager
- ETL = Extract Transform Load (data warehousing)
- MVC = Model View Controller (architecture to develop UIs)
- Various Microsoft/Apple/Google etc. terms (i.e. Windows SCCM = System Center Configuration Manager)

Out of the Box: Public Sector

- Say you want a government related job dealing with cybersecurity
- Obvious choices: NSA, CIA, FBI, etc.
- Semi obvious choices: ATF, DEA, DIA, SIGINT, DARPA, etc.
- Keep in mind: ALL GOV OFFICES NEED IA
- i.e. I worked for a subsidiary of the Department of the Interior
- Don't rule out the military industrial complex: Raytheon, Aero Company, Quasar Federal Systems (EM), Academi a.k.a. Blackwater, etc.

Out of the Box: Private Sector

- Name companies that you think are currently advertising to fill data warehouse positions in PA right now.
- Think deeply about it……
- Obvious answers: Deloitte, Highmark, PNC, etc.
- Not so obvious answers: Giant Eagle and Sheetz!

Resources

- The IUP career center
- Stack Overflow Careers and other job search sites
- MSN Careers and other sites featuring guides for different careers
- Learn the 12 steps of the Joel Test that test the quality of a software development team
- Social networking (think LinkedIn and actually talking to people)

Western PA (Pittsburgh)

- UPMC
- Google Pittsburgh
- PNC
- Alcoa
- USS
- CSC
- RAND Corporation
- Universities (CMU's CERT, Pitt, etc.)
- PMC
- Highmark
- IBM
- Phillips Global
- Giant Eagle
- Heinz
- GE
- Pittsburgh Glass Works
- YinzCam
- Grant Street Group

Central PA (Harrisburg)

- Penn State University
- Verizon
- PinnacleHealth
- JFC Staffing
- JR Associates
- TE Connectivity
- Astraya Corporation
- Tait Towers
- Advanced Technology Solutions
- Orbis
- CEG Partners
- CALNET Inc.
- Novitas Solutions
- AGI
- TecPort Solutions

Eastern PA (Philadelphia)

- Johnson & Johnson
- Siemens
- Bank of America
- BlackRock
- Universities (UPenn, Drexel, etc.)
- JP Morgan
- FBI/local Police departments
- McKean Defense Group
- BAE Systems
- Ebay
- ORACLE
- ARRIS
- Epicor
- Cigna
- Stroll
- iCorps Technologies
- QVC
- Pearson
- Comcast

NSA Recommended Best Practices and the NSA's Hackers

By Collin Donaldson

Road Map

- Password Generation and Management
- Safe Social Media
- Identity Theft Protection
- Securing Your Private Network
- Operations Security
- The NSA's Hackers: The TAO Unit

Password Generation and Management

- You all have been told dozens of times the importance of choosing strong unique passwords
- Passwords that are
  - Long (12+ characters)
  - Unique (not used elsewhere)
  - Contain numbers, lower/uppercase letters, and special symbols
  - Do not contain words
- However, there are other considerations to be made

Password Generation and Management

- Another consideration is using non-password access control techniques (biometrics, smart-cards, etc.) appropriately.
- Biometrics can be effective but discretion is required in choosing which one to use, as they vary wildly in security/usability
- Likewise, other methods used often at workplaces and for mobile devices such as smart cards, PINs, and pattern unlocks have their pros and cons.
- Another important consideration is whether or not to use software like Dashlane to generate/manage passwords for you

Safely Using Social Media

- Social media poses a major threat to both information security for individuals and organizations alike
- Some of the threats of using social media include:
  - Identity Theft (described later)
  - Loss of employment/ Loss of opportunity to gain employment
  - Defamation
  - Cyberbullying/extortion
  - Stalking
  - Doxing
  - Criminal charges pressed

Safely Using Social Media

- Again, everyone has probably heard the “dos and don’ts” of social media, but some are more important
- Posting information, particularly Personally Identifiable Information (PII), opens you up to unnecessary threats, so limit how much you post
- Another idea is to purposely falsify information
- Remember: anything you post can be used against you in the court of law, even if you are a victim

Safely Using Social Media

- Probably the single most important step to safely use social media is to keep a high level of vigilance regardless of the social media platform and its context
- DO NOT assume what you post is private or anonymous
- Enable privacy settings for all the social media accounts you use (can be partially automated with 3rd party software like AVG privacy fix)
- Social media platforms like Snapchat, Yik Yak, and Fade: Nothing Lasts Forever all claim to provide various degrees of anonymity, whether that is true is questionable

Identity Theft Protection

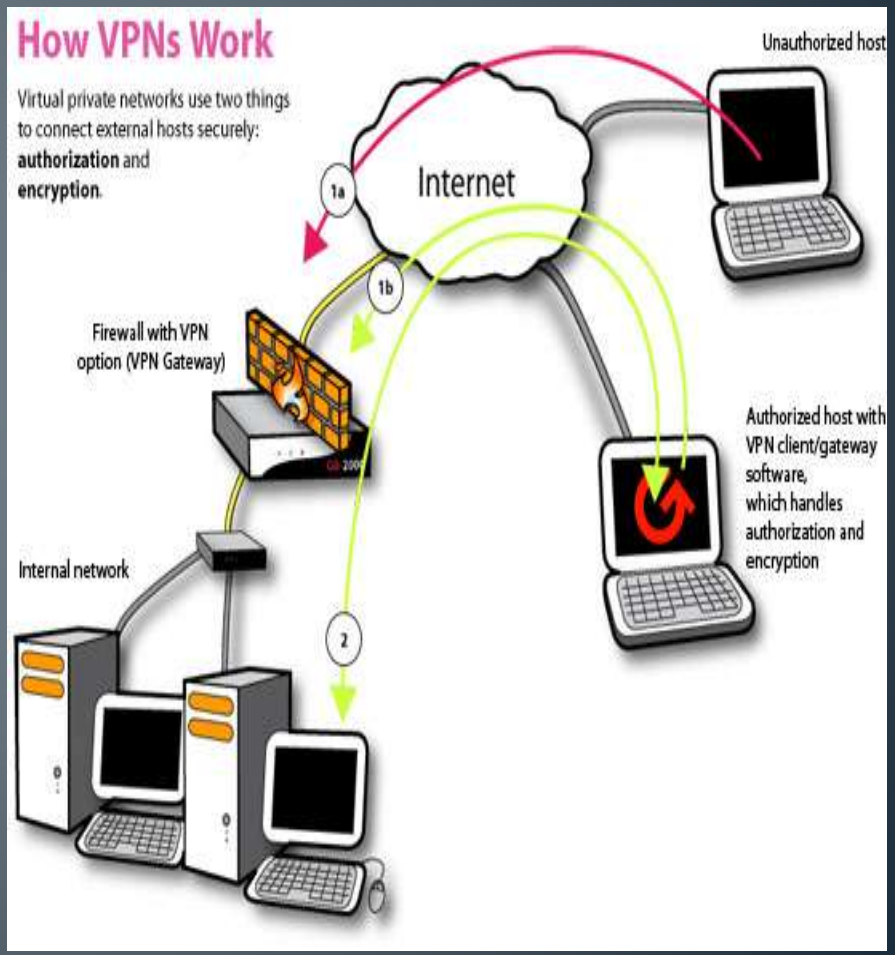
- As outlined in the previous counter-surveillance presentation, information brokers can discover and share a lot of information about you just from what you put on the internet
- This makes it much easier for identity thieves to steal your identity
- There are two main ways to avoid this:
  - Follow the steps outlined in the counter-surveillance presentation (namely unsubscribe from services that log your info)
  - Subscribe for identity theft protection, or simply keep a solid track on your finances and other transactions

Securing Your Private Networks

- It is worthwhile to reiterate private network security methods, even though many of them are common InfoSec knowledge.
- Steps to take include:
- Replace the default password with a strong password of your own
- Do not broadcast your SSID
- Ensure your network and all endpoint devices are encrypted via WPA2 if possible
- MAC filtering is also advisable for adding a further layer of security

Securing Your Private Network

- Some other advice:
- Disable remote administration
- Enable strict NAT
- Close any unnecessary ports on your router
- Use a VPN or onion routing for extra security



Operations Security

- OpSec is: the process of identifying critical information and determining if it is observable and useful by adversaries.
- The idea is to protect little pieces of data that could be grouped together by an adversary to reveal a bigger picture
  - "Loose lips sink ships"
- Not to be confused with operationAL security
- Related to InfoSec, Communications Security (COMSEC), Signal Security (SIGSEC), Transmission Security (TRANSEC), and counter-intelligence.
- Originally a military term aimed at opposing armies, the term has expanded to civilian use in reference to hackers, industrial espionage, law enforcement, and mass surveillance.

Operations Security

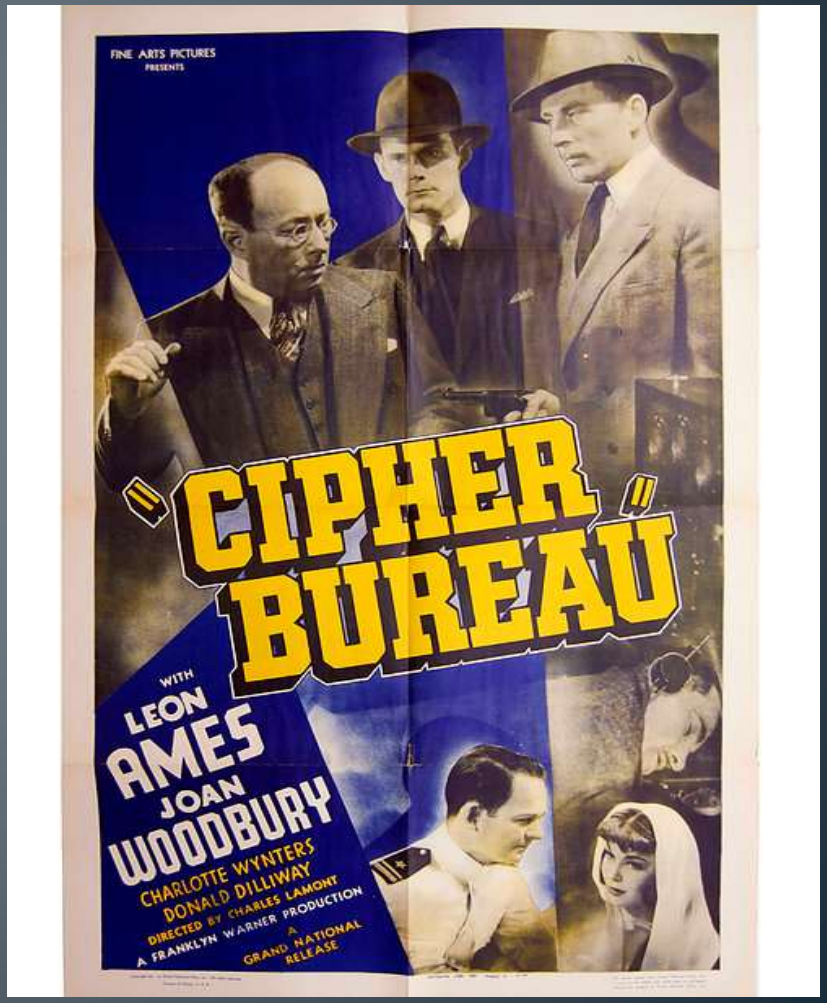
- OpSec can be considered a subset of risk management aiming at keeping information confidential
- It includes identifying four phases: critical information, threat analysis, vulnerability analysis, and countermeasures
- Phase 1: What is critical to you or your organization? What information can lead to clues about said information?
- Phase 2: Who is the adversary and what are their capabilities? Who wants or needs our information? Is the threat active, passive, or inadvertent?
- Phase 3: How vulnerable is the information? What can we do to make it more secure?
- Phase 4: Eliminate/reduce vulnerabilities, disrupt information collection, or prevent accurate information interpretation

Operations Security in InfoSec Example

- Situation: Company employees are attending an important conference to showcase company technology
- Phase 1: Trade secrets concerning the technology to be showcased such as specifications, blueprints, and emails about the project
- Phase 2: Rival companies, domestic/foreign intelligence, disgruntled insider, malicious hackers
- Phase 3: Employees are using public Wi-Fi, posting about the conference on social media, and information regarding the technology (i.e. emails) are not encrypted
- Phase 4: Managers make employees use a company VPN, moderate social media and ask employees to refrain from posting anything, and all critical information is encrypted

The NSA's Hackers: The TAO Unit

- The NSA has been involved with national security regarding computers and signal intelligence (SIGINT) since its inception in 1952, and even before then by its predecessor The Cipher Bureau (a.k.a. The Black Chamber)
- Since Edward Snowden's leaks, the organization has become synonymous with mass surveillance and hacking
- However, it is the activities of the NSA's Tailored Access Operations (TAO) unit people are actually referring to

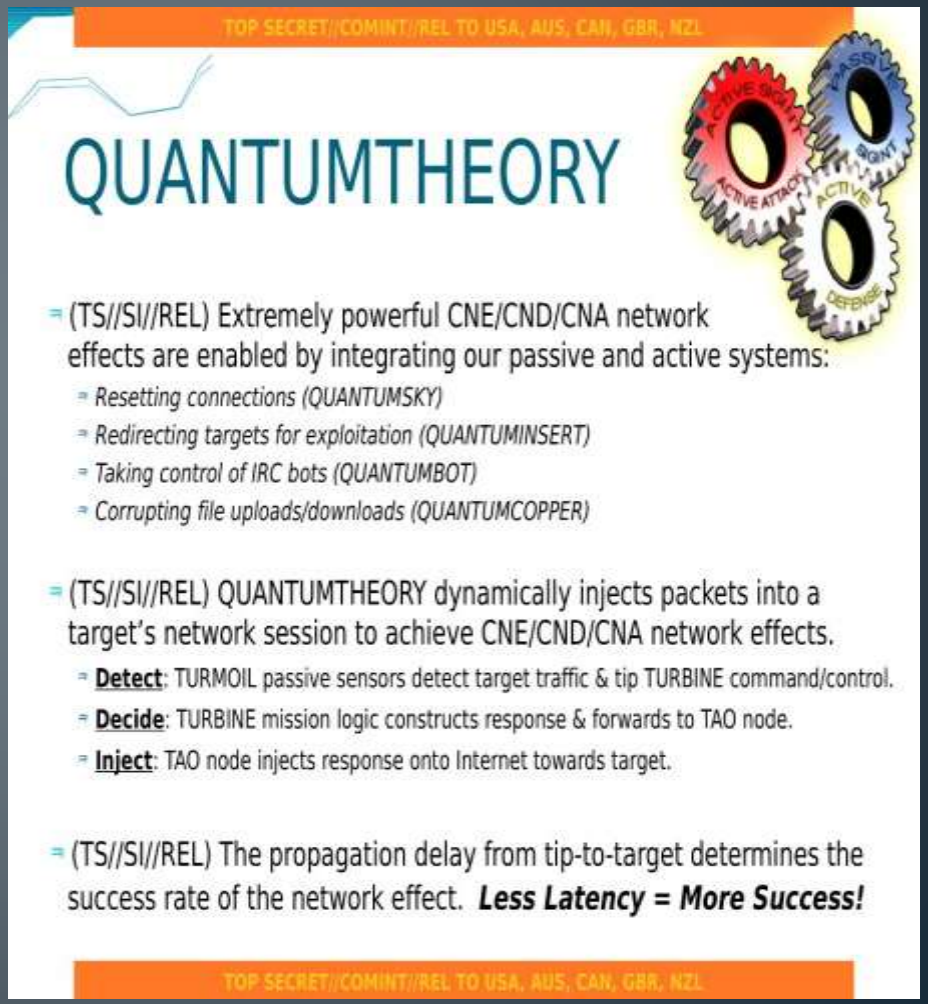


The NSA's Hackers: The TAO Unit

- The TAO unit is the centerpiece of the NSA's SIGINT operations, of over a 1,000 military and civilian employees and assisted by the CIA, FBI, Britain's GCHQ, and major companies such as Sprint and Microsoft
- Some of TAO's known targets include:
  - China
  - OPEC
  - Mexico's Secretariat of Public Security
  - Tor/Firefox users
  - International GRX (tele communications) provides like Belgacom
  - Optical submarine fibers and fiber providers

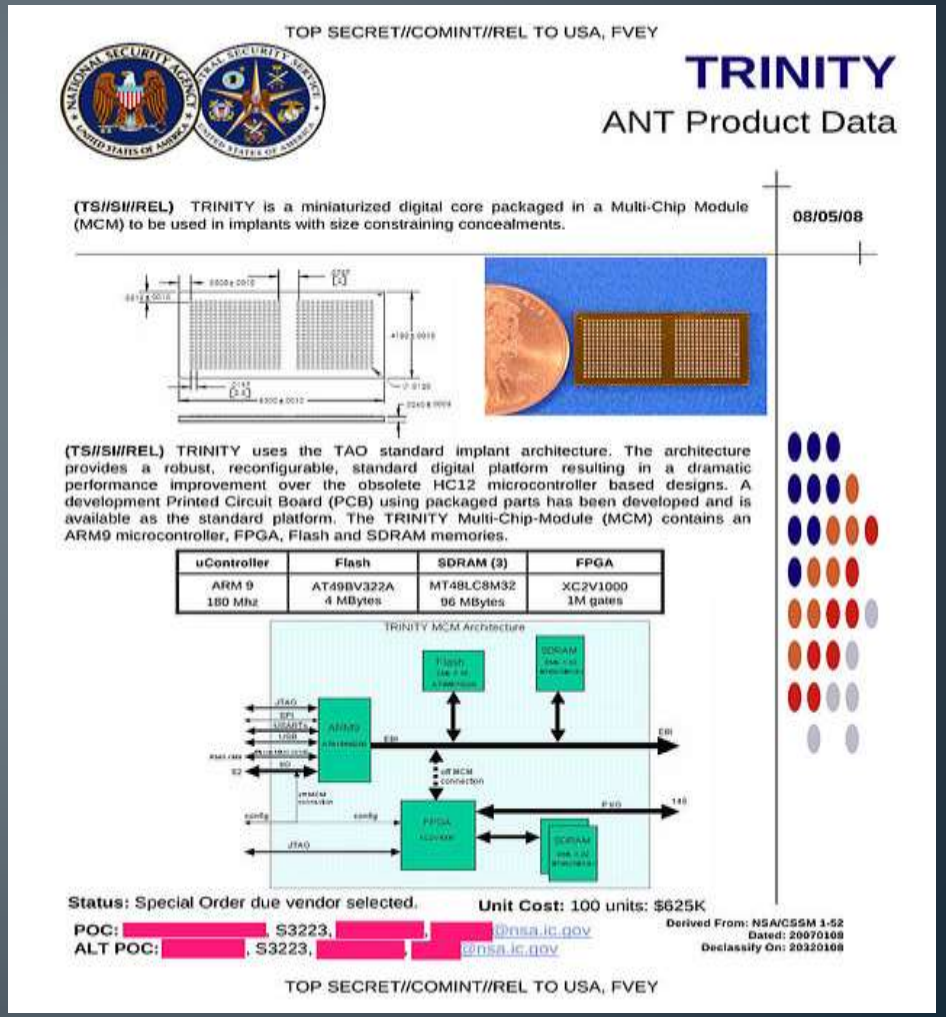
The NSA's Hackers: The TAO Unit

- Some TAO attack vectors include:
  - Intercepting purchased computers and installing them with spyware/hardware
  - Installing backdoors in company servers and software/hardware products
  - Spoofing telecommunications networks
  - The ANT catalog showcases a variety of R&D developed software and hardware attack vectors



The NSA's Hackers: The TAO Unit

- Some TAO codenames:
- QUANTUM attack suite
- XKeyScore : Vulnerability database
- FOXACID: Browser exploits
- CANDYGRAM: Cell tower spoofer
- TOTEGHOSTLY: Win-phone Remote Control
- RAGEMASTER: VGA signal interceptor
- COTTONMOUTH: Compromised USB/Ethernet
- HOWLERMONKEY: RF transmitter
- SOMBERKNAVE: Remote Access
- GENESIS: Phone bug
- IRATEMONK: Firmware exploiter



TAO Video

- https://www.youtube.com/watch?v=pOMBZPOligA

By Collin Donaldson

What is it?

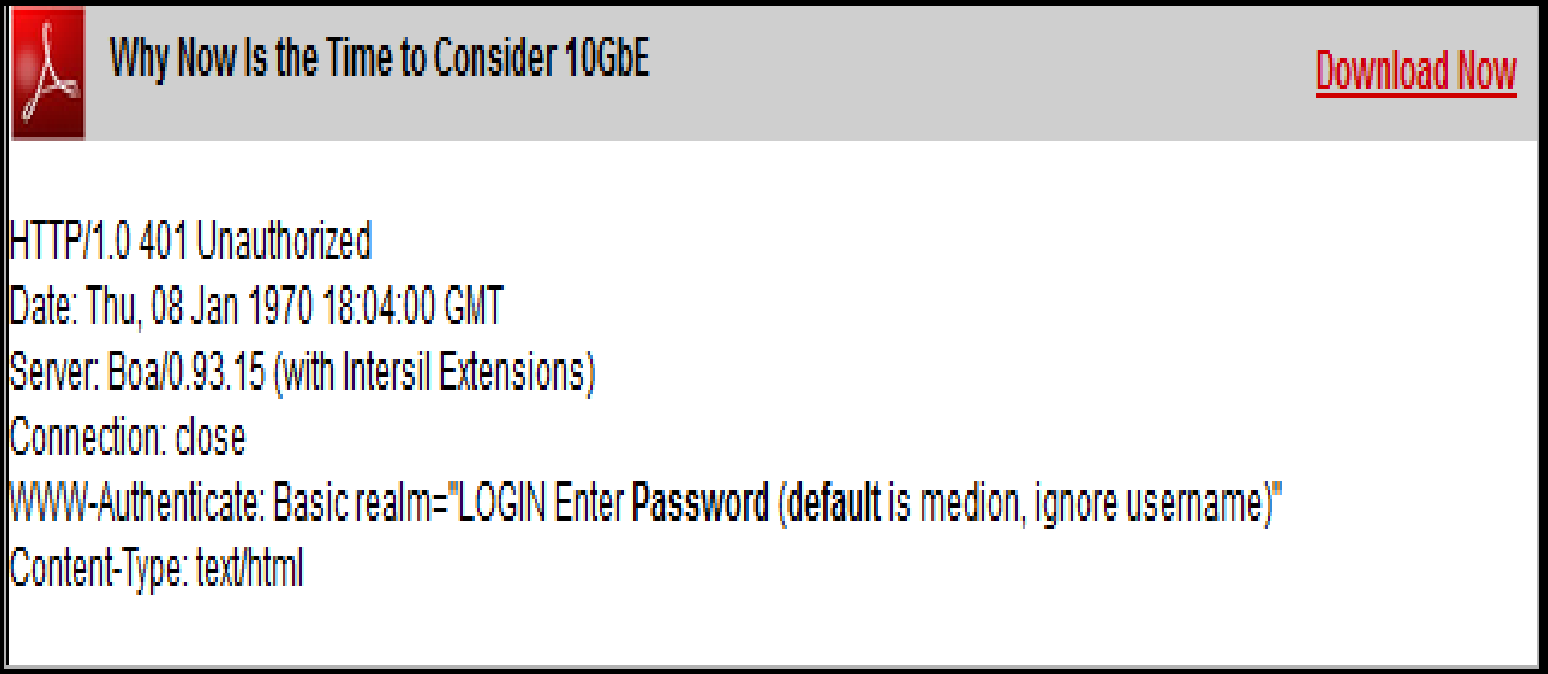
- Shodan is a search engine that allows you to look for devices connected to the internet using service banners.
- When you connect to a server listening on a given port, the server (usually) responds with a service banner.
- Service Banner: A block of text about the given service being performed.

- Shodan uses a technique called “Banner Grabbing”
- Banner Grabbing is an enumeration technique used to glean information about computer systems on a network and the services running its open ports.
- Indexes banners instead of web content
- Admins can use it to keep tabs on the services and systems on their networks
- Hackers can use it to expose potential targets

The How

Potential Targets

- Routers
- Webcams
- SCADA systems
- Traffic Lights
- Note: Be careful what you try to access!



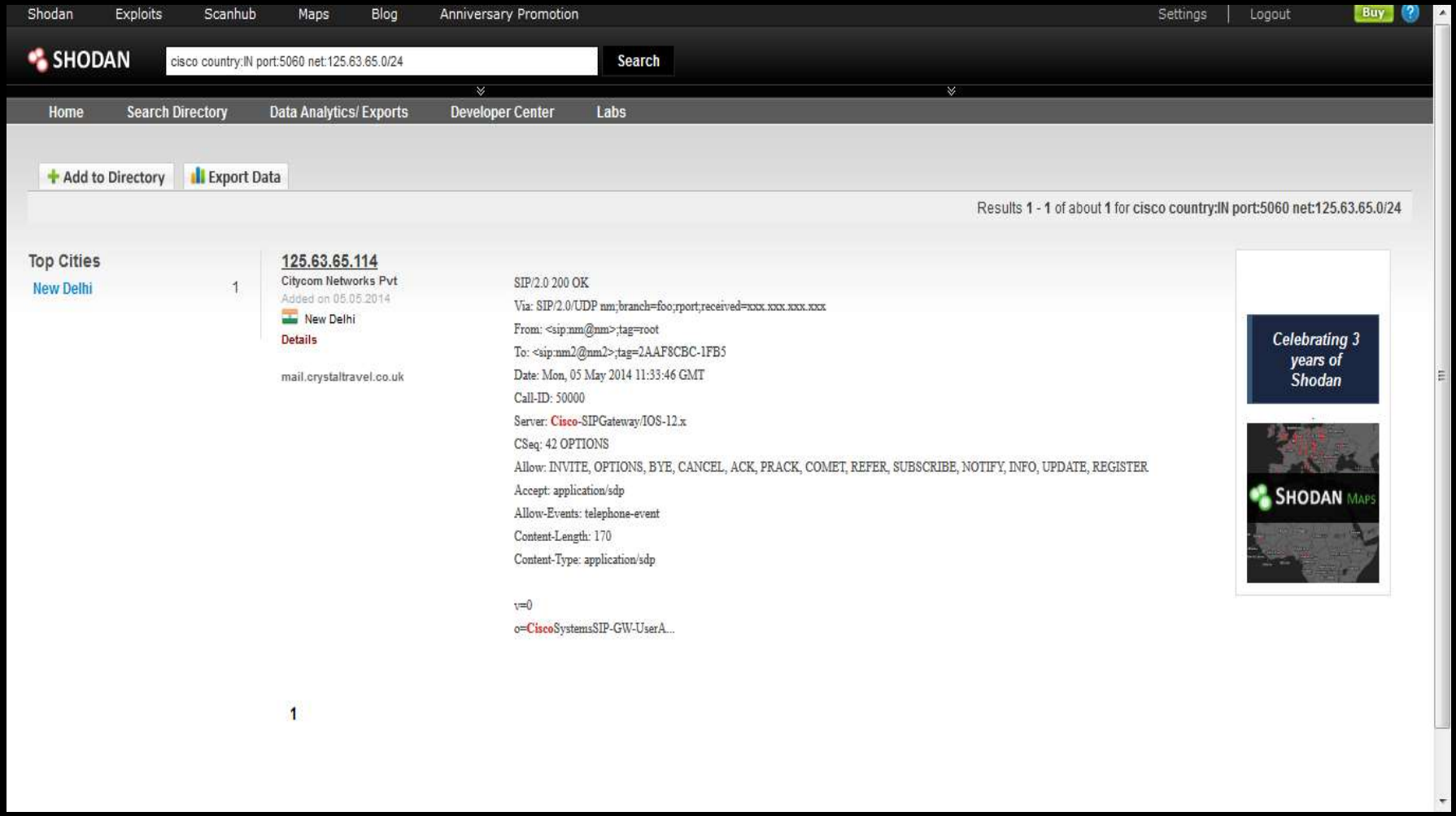
We now have the HTTP, Server (Boa is a lightweight server for embedded systems such as Androids), and the default password.

Service Banner Example

Getting Started: Create an Account

Familiarize Yourself

- Shodan has similar features and functionality to other search engines, but the searches are quite different
- Check out “popular searches” for some starting tips
- You can filter by banner type, port, OS, country, latitude/longitude, etc.
- Example: cisco country:IN port:5060 net:125.63.65.0/24
- Result on next Slide



Citycom Networks Pvt, New Delhi, India

Some Useful Search Terms

- Use *net:your.ip.add.ress* o r *net:your.ip.add.0/24* to pen-test your own network
- "iis/5.0“ for Internet Information Services
- Network/Company Type names (“Cisco”, “Apache”, “Telnet”, etc.)
- Search software for your type of target (i.e. “webcamxp” is common webcam software).

User Authentication

- Some servers require authentication
- Use lists of common default usernames and passwords such as http://www.phenoelit.org/dpl/dpl.html
- You could also use more advanced tools like Cain and Abel if you really want to break a password.


Getting The Most Out Of Your Smartphone
By Collin Donaldson

Conventional vs Phone/Tablet Architecture

- In conventional OS architectures, the user accesses the OS via an account that has certain privileges (admin, guest). They can then interact with applications as much as their account status allows them.
- Phone/Tablet OS architectures treat applications as individual users requiring their own permissions. The underlying human user is stuck with guest-level privileges.

Why Are You A Permanent Guest?

- Safety/Security: You lack “write” access so you cannot accidentally trash your system or install unscrupulous applications.
- The OS provider (Google/Apple/Microsoft, etc.) can force you to use their official app store and restrict you from uninstalling their default apps.
- Your carrier can restrict you from uninstalling their default apps, changing their version of the OS (if they modified it), and “lock” you from using other carriers.

The Solution?

- Gain "Admin" account level privileges!
- This lets you:
  - Install your own software/firmware
  - Access apps and app features you couldn't before
  - Uninstall default apps
  - Unlock your phone to use other carriers
  - Access hardware options (overclock/underclock the processor)
  - And more!

Terminology

- Rooting: Process that enables Superuser account (root) on your Android device.
  - Used as a generic term for gaining Superuser/Root/Admin account status.
  - Derived from Linux, which Android is partially based on.
- Jailbreaking: Rooting an Apple iOS device.
  - Apple places additional restrictions on rooting such as locking the bootloader, so the term is used for iOS rooting.
- Unlocking: Usually refers to bypassing carrier restrictions.
- Tethering: Using your phone as a modem (i.e. using your phone as a Wi-Fi hotspot)

More Terminology

- Flashing: Installing different firmware or a custom-ROM onto your smartphone.
- Custom-ROM: Standalone version of the OS that has been customized by someone.
  - Typically ROM-images are in .zip or .tar format.
  - Similar to a recovery image or a game emulator.
- Sideloading: Downloading/Installing data on to your phone via methods outside of the official Google/Apple/Microsoft stores.
  - Examples: USB, Bluetooth, Wi-Fi, SD Card, etc.

Disclaimer!

- There are some dangers of rooting your phone.
- Incorrectly rooting your phone can “brick” it.
- Backup your phone first, rooting will wipe it!
- Voids your phone’s warranty (and some app warranties)
- Your phone OS is probably not designed to be accessed via root, so it may become more unstable (buggy).
- Opens your phone up to more malware if you aren’t careful.
- Luckily, rooting is reversible via “unrooting”.

Before We Begin

- Exact steps differ depending on OS, Brand, and Model, yet the process is roughly similar for all.
- We will be using established root methods, not rooting a never-before rooted device.
- There are also more automated services out there like OneClickRoot.
- I will briefly explain the steps to manually root a Samsung Galaxy S4 and jailbreak an iPhone 6 manually.
- Required: a phone (duh), a PC, a USB cable, an SD card, and 3rd party software.

Before We Begin (cont)

- This will not cover how to flash a Custom-ROM or any other software on to your device
- However, a useful thing to know for both rooting and flashing is how to get into your phone's BIOS menu.
- Typically, while the power is off, hold power + volume down. Varies depending on device.
- There are many rootable devices, and often different approach for different variables (i.e. PC option, Mac option, etc.)

Rooting the Samsung Galaxy S4 (Android)

1. Download and install Samsung USB Drivers on your computer.
2. Go to Settings > About Device. Tap “About Device” 7 times to enable Developer options (on older Android versions, this is not necessary).
3. Go to Developer Options and enable USB Debugging
4. Plug your device into a PC via a USB cable
5. Download MotoChopper on your PC

Galaxy S4 Root (continued)

6. Extract the .zip file to your desktop

7. Double click the “run .bat” file inside the extracted folder.

8. When prompted “Allow USB Debugging?” on your phone, press “OK”.

9. In your applications drawer of your phone, check for an app called “Superuser”. If present, your phone is rooted.

Jailbreaking the Apple iPhone 6 (iOS)

1. Download Pangu Jailbreak to your computer
2. Connect your device to your computer using a USB cable
3. Enter Airplane mode, Disable Passcode from Settings > Touch ID & Passcode and turn off Find my iPhone from Settings > iCloud > Find my iPhone.
4. Launch the Pangu.exe file as an administrator (right-click, run as administrator)

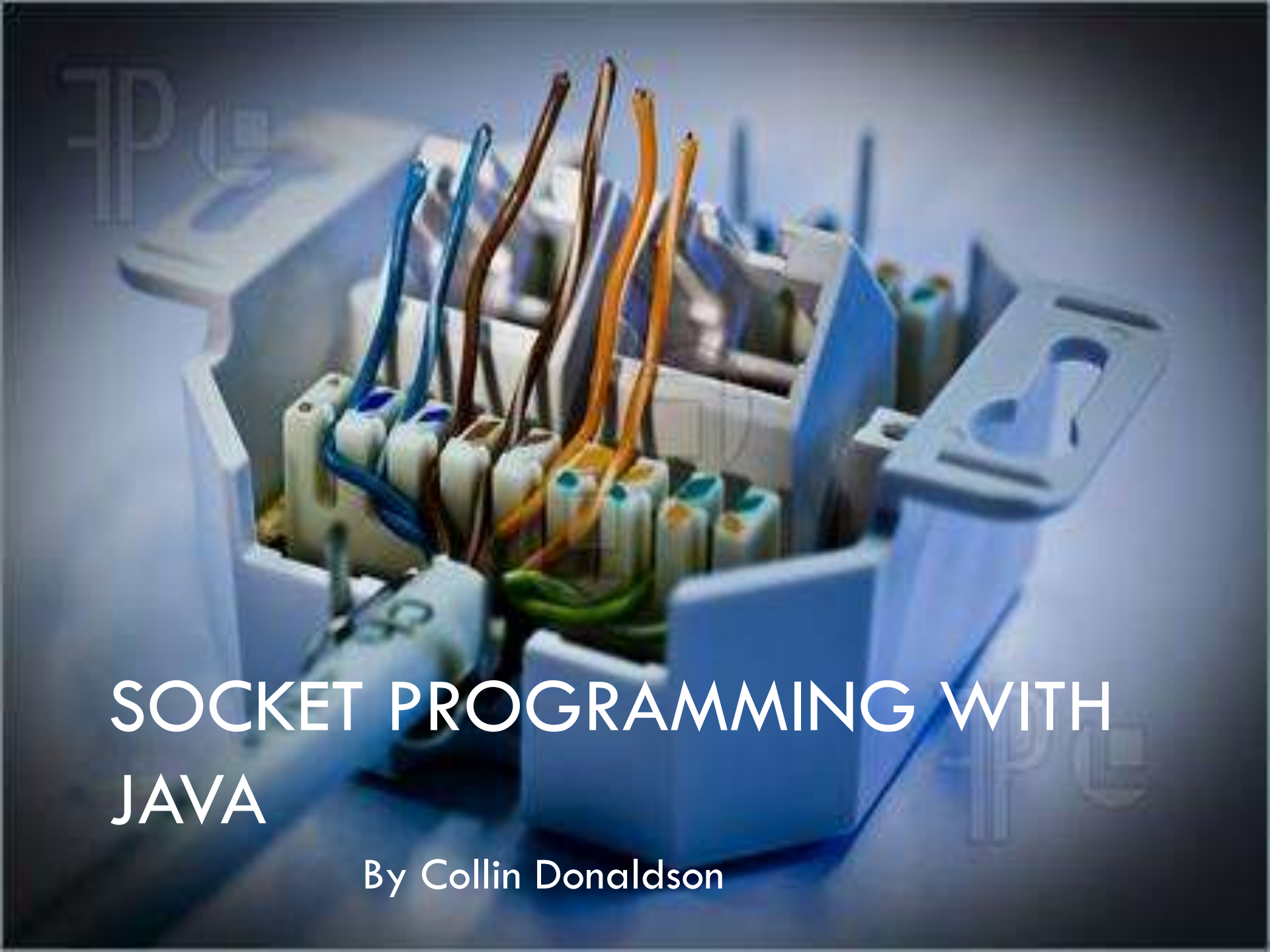
iPhone 6 Jailbreak (cont)

5. Wait for Pangu to detect your device, once it has click "Start Jailbreak".

6. Click "Already did" on the next screen.

7. Wait for the progress bar to complete. Your iPhone will reboot.

8. Once completed, you should see the Cydia icon on your home screen. If you see it, the jailbreak was sucessful.



SOCKET PROGRAMMING WITH JAVA

By Collin Donaldson

Definitions

- Network Socket: An endpoint of a two-way communication link between two programs running on a network.
- Based on the Open-Read-Write-Close format first used in UNIX.
- Socket Address: IP address + port number. Used so programs can identify connections and send packets to the appropriate process/thread
- Similar to how one end of a telephone connection is identified by a phone number + extension.

Socket Life Cycle

- Open-Read-Write-Close
- Creation (Open Socket)
- Reading and Writing (Receive and Send to Socket)
- Destruction (Close Socket)

Socket Communication Protocols

- Datagram Communication: User Diagram Protocol (UDP). A connectionless meaning that each time you send datagrams, you also need to send the local socket descriptor and the receiving socket's address. Additional data must be sent each time a communication is made.

Socket Programming Protocols (cont)

- Stream Communication: Transfer Control Protocol (TCP). TCP is a connection-oriented protocol. In order to communicate over the TCP protocol, a connection must first be established between the pair of sockets. While one of the sockets listens for a connection request (server), the other asks for a connection (client). Once two sockets have been connected, they can be used to transmit data in both (or either one of the) directions.

Which is better for Client/Server Apps?

UDP (Datagram)

- No Connection Setup Time
- 64 kilobyte limit per location
- Unreliable (packets could arrive out of order)

TCP (Stream)

- Connection Setup Time
- No limit
- Reliable (packets always arrive in order)

Connection Setup Time

In Summary

- TCP is useful for implementing network services , such as remote login (rlogin, telnet) and file transfer (FTP), which require data of indefinite length to be transferred.
- UDP is less complex and incurs fewer overheads. It is best for implementing client/server applications in distributed systems built over local area networks (LANs).

On to the tutorial!

- Note! This tutorial only covers Stream/TCP since it is more common.
- Start a new project and import the following
- import java.io.*;
- import java.net.*;
- Note: We will be using the client-server model. Processes are partitioned between providers (servers) and requesters (clients)

Open Socket from Client

- Socket MyClient;
- try { MyClient = new Socket("Machine name", PortNumber); }
- catch (IOException e) { System.out.println(e); }
- When choosing port number, use on that is above 1,023. Numbers below 1,023 are reserved for privileged users (root/super user) and standard services. For example, HTTP is port number 80 and HTTPS is port number 443.

Open Socket from Server

- ServerSocket MyService;
- try {
- (MyServerice = new ServerSocket(PortNumber);
- }
- catch (IOException e) {
- System.out.println(e); }

Server side Socket Object that listens for and accepts connections from other users

- Socket clientSocket = null;
- try {
- serviceSocket = MyService.accept();
- }
- catch (IOException e)
- { System.out.println(e);
- }

Client side Input Stream

- DataInputStream input;
- try {
- input = newDataInputStream(MyClient.getInputStream());
- } catch (IOException e)
- {
- System.out.println(e);
- }

Sever Side Input Stream

- DataInputStream input;
- try { input = new DataInputStream(serviceSocket.getInputStream());
- }
- catch (IOException e) {
- System.out.println(e);
- }

Client side Output Stream using Data/PrintStream

- DataOutputStream output;
- try { output = new DataOutputStream(MyClient.getOutputStream()); }
- catch (IOException e) {
- System.out.println(e);
- }
- DataStream = primitives PrintStream = text

Server Side Output Stream using Data/PrintStream

- PrintStream output;
- try {
- output = new PrintStream(serviceSocket.getOutputStream());
- } catch (IOException e) {
- System.out.println(e); }

Closing Sockets Client Side

```
try {
output.close();
 input.close();
 MyClient.close();
}
catch (IOException e)
 {
 System.out.println(e);
}
```

Closing Sockets Server Side

- try {
- output.close();
- input.close();
- serviceSocket.close();
- MyService.close();
- }
- catch (IOException e) {
- System.out.println(e);
- }

Full Examples in Eclipse!

- Examples include:
- 1. Simple Mail Transfer Protocol (SMTP) client
- 2. Echo Server (takes input, echoes it straight back as output to one client along one thread).
- Note: It is likely that by default the SMTP will fail to recognize the host and the Echo Server will be blocked by your firewall
- http://www.javaworld.com/article/2077322/core-java/sockets-programming-in-java-a-tutorial.html?page=2


KALI LINUX
KALI | LINUX 1
By Collin Donaldson
the quieter you become, the more you are able to hear

Roadmap

- Origin of an OS: From UNIX to Kali Linux
- Linux Architectures and Ubiquity
- What is Kali and why should I use it?
- Kali's New Features
- Kali's Toolkit
- Installing Kali
- Summary

UNIX to Kali Timeline

- 1968: E.W Dijkstra develops MULTICS (Multiplexed Information and Computing Service) in the Netherlands

- 1969: Bell telephone (AT&T) lab researcher Ken Thompson developed a new system using MULTICS as part of a team. His coworker Brian Kernighan dubbed it UNICS (UNiplexed Information and Computing Service). It was later changed to UNIX.

- Milestone: The UNIX operating system was born.

UNIX to Kali Timeline

- 1969-1973: Bell Telephone researcher Dennis Richie develops the C language as a systems programming language for UNIX.
- 1970s: UNIX versions 6 and 7 were developed, first in B and Assembly than C. Originally for academic use, later sold to vendors.
- 1987: A Unix-like system based on microkernel design known as MINIX was developed.

- Milestone: C language developed.

UNIX to Kali Timeline

- 1980s-1990s: The “UNIX Wars” occur, vendors struggle to standardize UNIX.
- 1991:Linus Torvalds developed a new operating system called Linux, which is similar to MINIX.
- 1990s-Today: Various UNIX and UNIX/Linux-like distributions are released, such as: GNU, OS X, Debian, and Ubuntu.

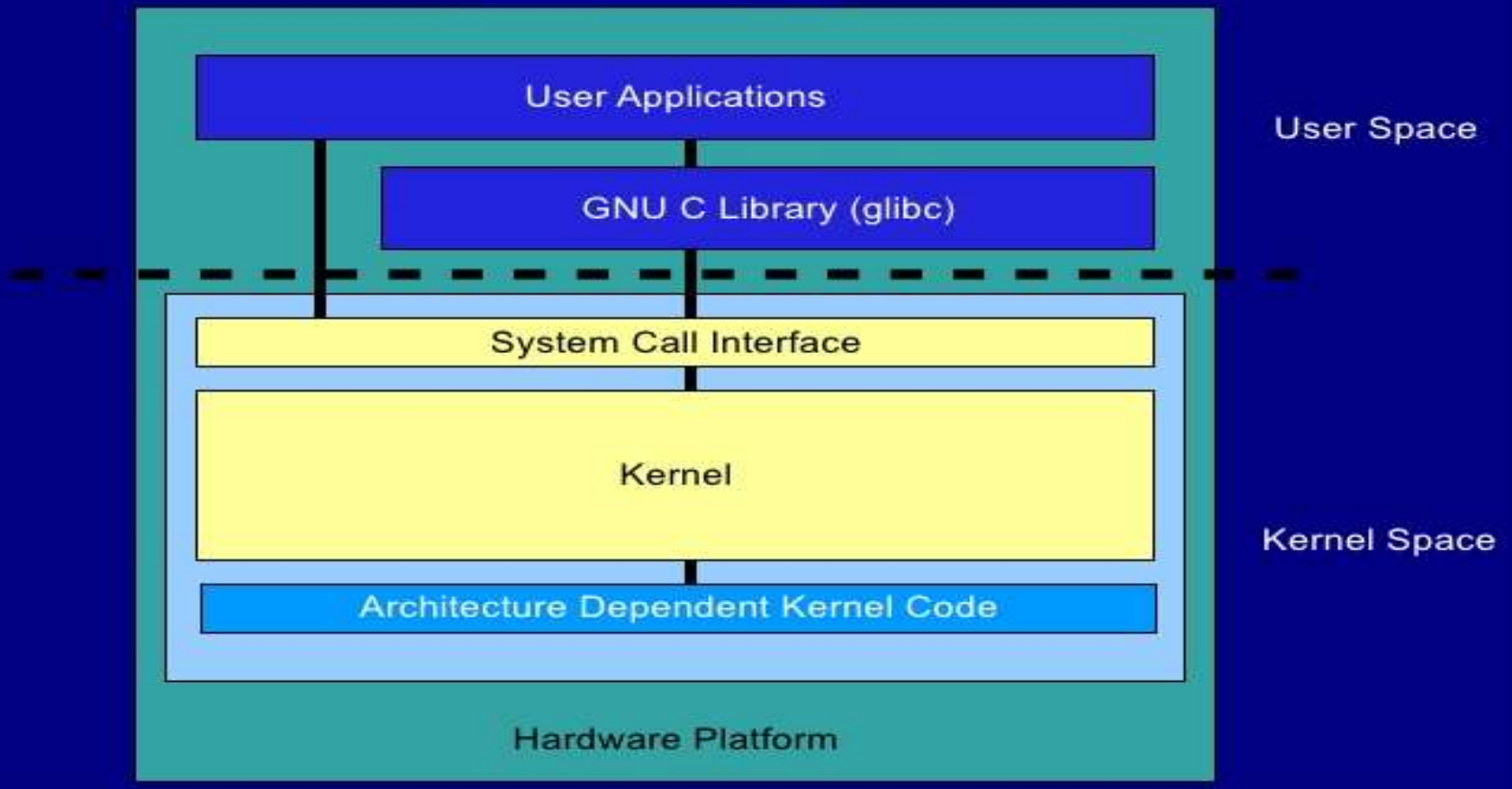
- Milestone: Linux was born.

UNIX to Kali History

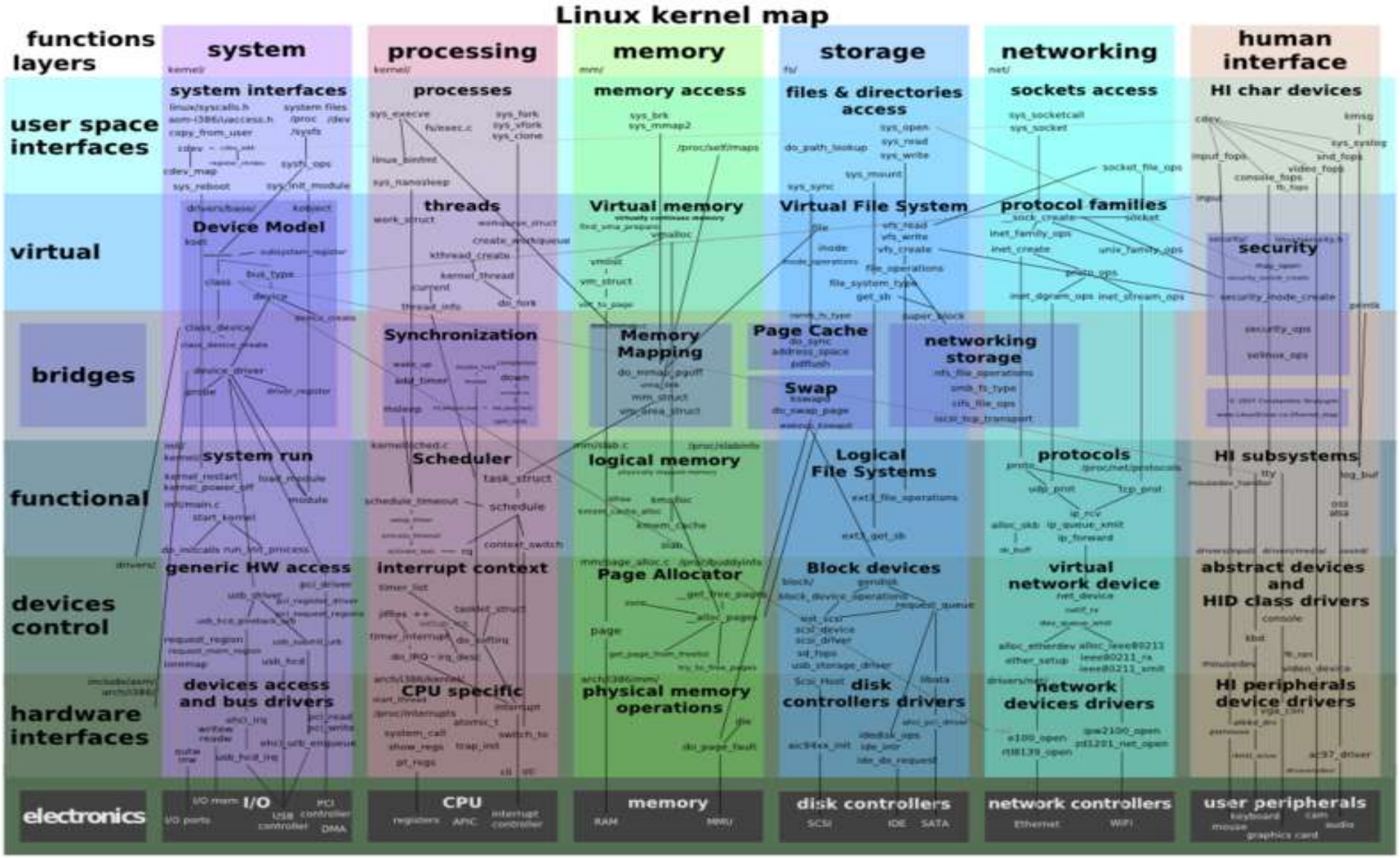
- 2006: Linux distribution BackTrack was released by Offensive Security. It becomes the definitive Penetration Testing platform available.
- 2012: The final BackTrack version, Backtrack 5 R3, is released.
- 2013: Kali Linux, a.k.a. BackTrack 6, is released by Offensive Security.
- Milestones: BackTrack is born, then Kali.

Basic Linux Kernel

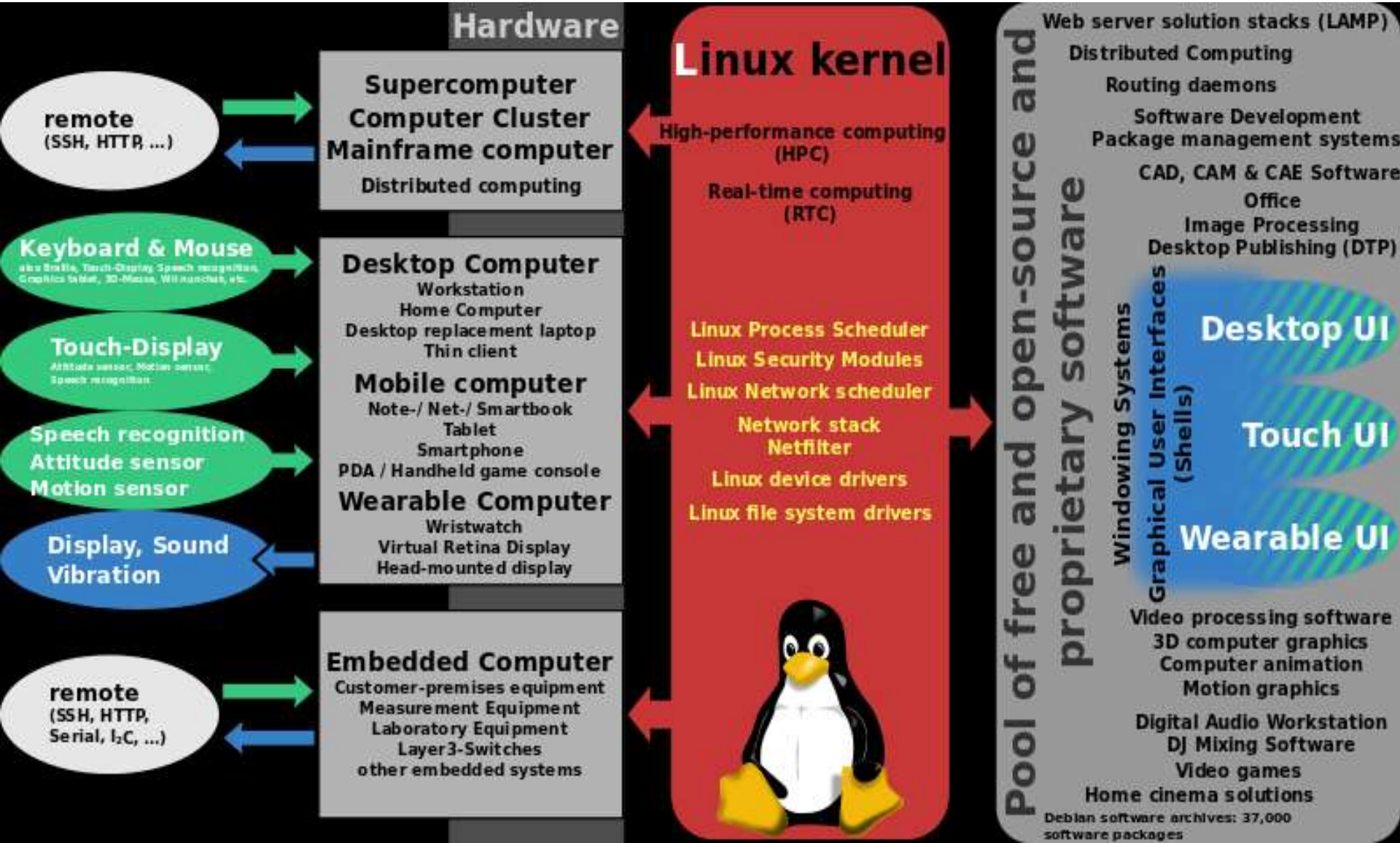
Fundamental Architecture



Full Linux Kernel



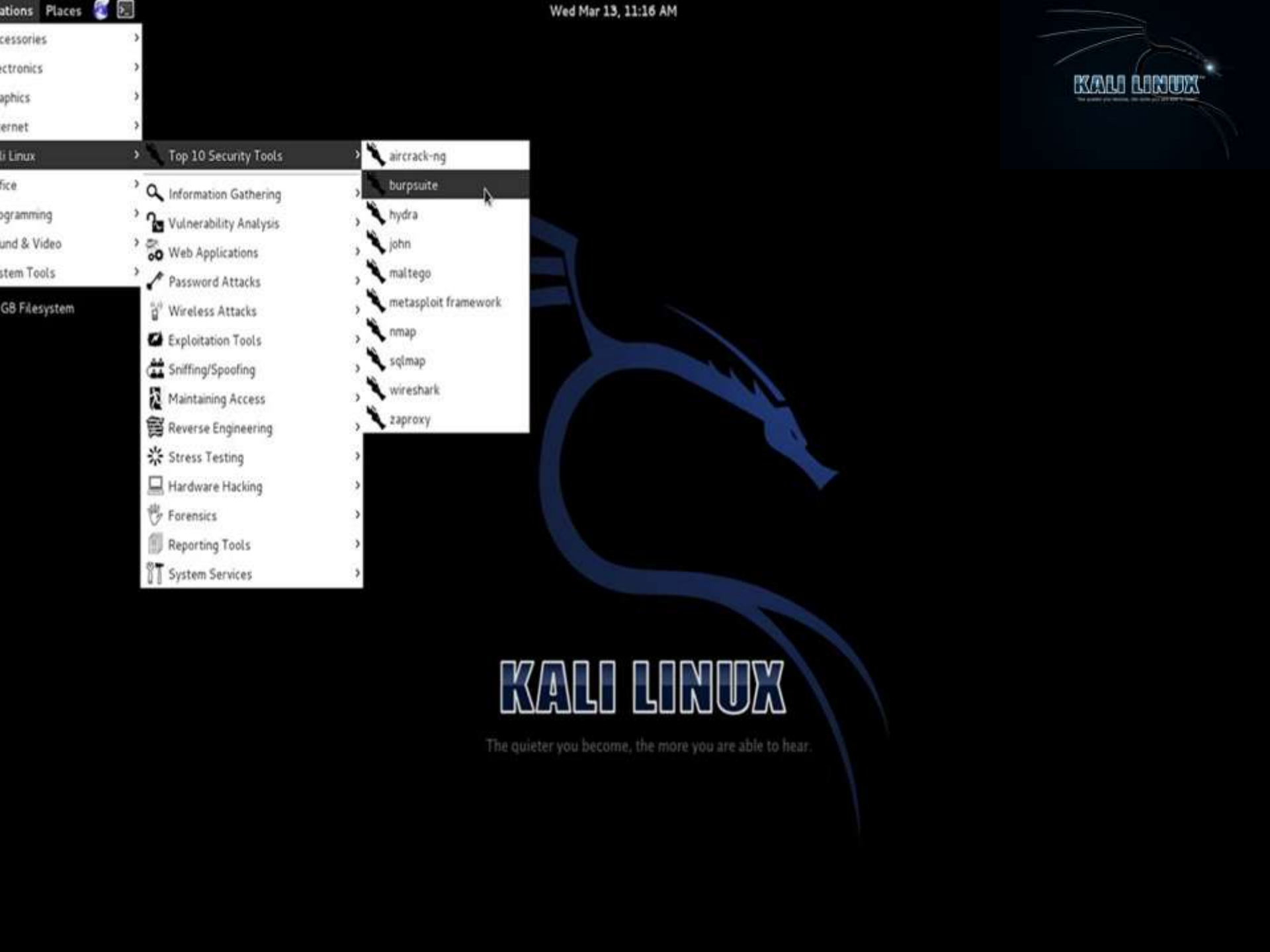
Linux Ubiquity



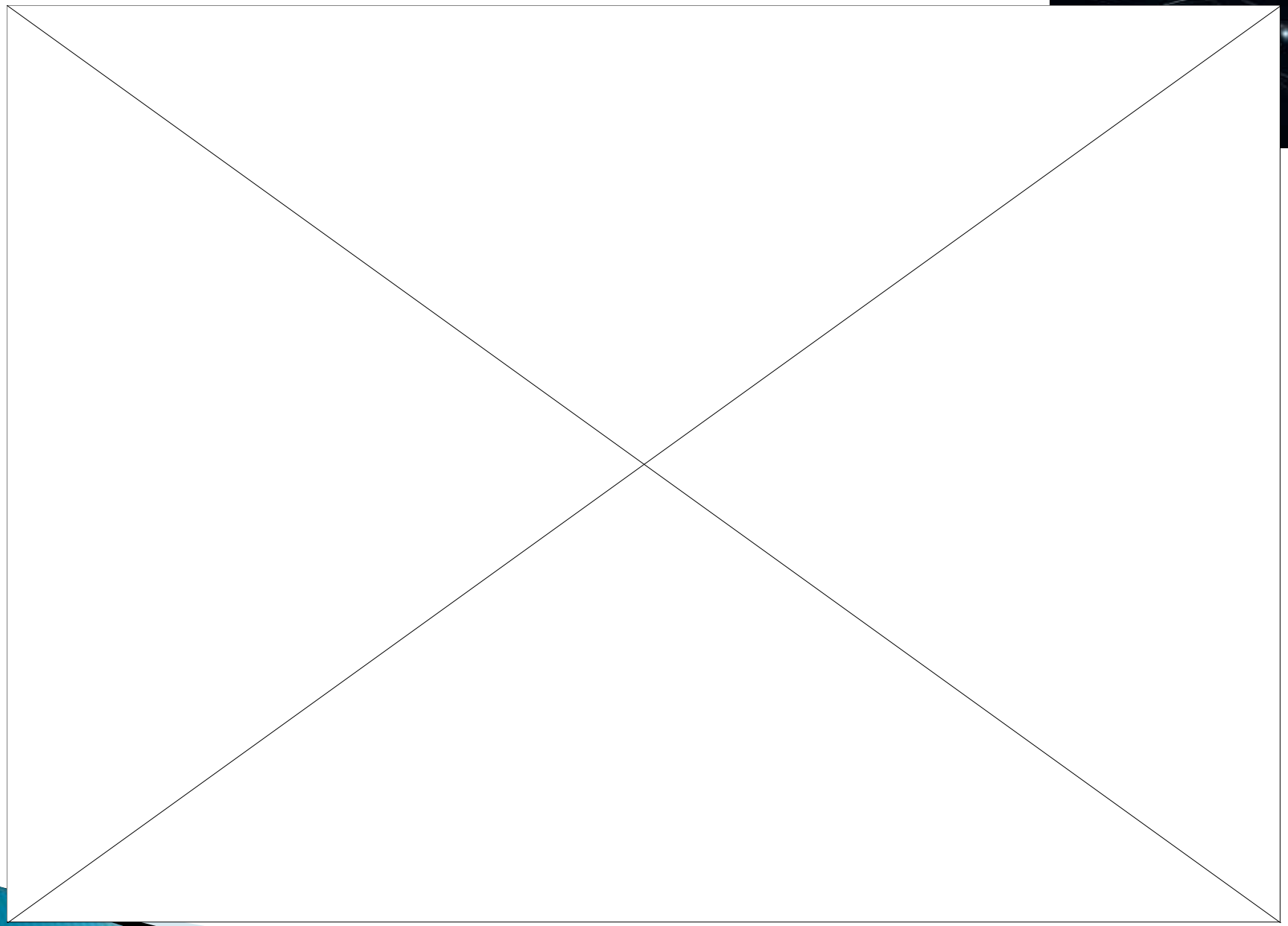
Get to Kali already!!!!

What is Kali Linux?

- Kali Linux is an advanced Penetration Testing and Security Auditing Linux distribution (distro). Named after a Hindu god.
- It was designed to replace the BackTrack Linux distro.
- A Linux distro is a operating system based off the Linux kernel.
- Think Windows NT and all the Windows distributions (XP, Vista, 7, 8, etc.)
- Linux is itself based off the UNIX kernel.
- UNIX > Linux > BackTrack > Kali.


ations
Places
Wed Mar 13, 11:16 AM
cessories
ctronics
aphics
ernet
li Linux
fice
ogramming
und & Video
stem Tools
GB Filesystem
Top 10 Security Tools
Information Gathering
Vulnerability Analysis
Web Applications
Password Attacks
Wireless Attacks
Exploitation Tools
Sniffing/Spoofing
Maintaining Access
Reverse Engineering
Stress Testing
Hardware Hacking
Forensics
Reporting Tools
System Services
aircrack-ng
burpsuite
hydra
john
maltego
metasploit framework
nmap
sqlmap
wireshark
zaproxy
KALI LINUX
The quieter you become, the more you are able to hear.


KALI LINUX
KALI LINUX
Boot menu
Live (amd64)
Live (amd64 failsafe)
Live (forensic mode)
Install
Graphical install
Advanced options
>
Press ENTER to boot or TAB to edit a menu entry



Why use Kali?

- It is FREE!!!!!
- 300+ applications, from password crackers to digital forensics software.
- Vast wireless device support (ARM processors).
- Completely Customizable.
- Multilingual Support.
- Secure Development Environment.
- Open source Git tree.
- Filesystem Hierarchy Standard (FHS) Compliant.
- Gnu Privacy Guard (GPG) secure signed packages and repos.

Kali's New Features

- Switched from Ubuntu to Debian.
- FHS and Debian compliant.
- Can now bootstrap and customize Kali ISOs.
- Streaming security and package updates from Debian.
- Expanded ARM development.

Kali's New Features (cont)

- Easy upgrade to future versions.
- Customizable desktop environment.
  Automatable Kali Installations.
- Streamlined Development Process.
- Long Term Tool Packaging and Maintenance.

Kali's Toolkit (300+!)

- Metasploit
- Nmap
- Wireshark
- Aircrack-ng
- John the Ripper
- CaseFile
- THC-Hydra
- Arduino
- diStorm3
- Sqlninja
- Proxy Strike
- Ghost Phisher
- CryptCat
- WebScarab
- Android-sdk
- Maskprocessor
- SIPArmyKnife
- FERN Wi-Fi Cracker

Installing Kali

- Burn to a live disk, insert into disk drive, install.
- Format a flash drive, install Kali Linux on it, insert into a computer, at BIOS screen select boot from USB.

Via Physical Media

- Download VMware or other similar software, create a virtual machine, download and install Kali Linux onto virtual machine.
- Use a Preboot eXecution Environment (PXE) to install and boot Kali Linux from a server/client network.

Via Digital Media

Summary

- Bell Telephone (AT&T)'s UNIX > Linus Torvalds's Linux > Offensive Security's Linux Distribution BackTrack > Kali Linux a.k.a. BackTrack 6.
- Kali is the premier operating system for Penetration Testing and other related uses.
- Kali was built from the ground up to replace BackTrack.
- There are many platforms and installation methods that are Kali-compatible.

Sources

- Admin. "Brief History of Unix and Linux Operating Systems." More Process, 18 Oct. 2013. Web. 23 Sept. 2014.
- Aharoni, Mati. "Kali Linux | Rebirth of BackTrack, the Penetration Testing Distribution." *Kali Linux*. Offensive Security, 25 Aug. 2014. Web. 23 Sept. 2014.
- Cimafranca, Dominque M. "Architecture Of The Linux Kernel." *Architecture Of The Linux Kernel*. Ateneo De Davao Universit, 13 Aug. 2009. Web. 23 Sept. 2014.
- “Backtrack vs Kali." *Diffen.com.* Diffen LLC, n.d. Web. 23 Sep 2014.
- "Linux vs Unix." *Diffen.com.* Diffen LLC, n.d. Web. 23 Sep 2014.
- Wikipedia. "BackTrack." *Wikipedia*. Wikimedia Foundation, 23 Sept. 2014. Web. 23 Sept. 2014.
- Wikipedia. "Linux." *Wikipedia*. Wikimedia Foundation, 23 Sept. 2014. Web. 23 Sept. 2014.
- Wikipedia. “UNIX." *Wikipedia*. Wikimedia Foundation, 23 Sept. 2014. Web. 23 Sept. 2014.

Link to install (VMware method)

- http://www.kalitutorials.net/2014/09/installing-kali-linux-on-android-via.html

VIRTUALIZATION TECHNOLOGIES

BY COLLIN DONALDSON

PHYSICAL COMPUTING

Install Hardware

Load Operating System and other software

Deploy either manually or via a network

To Run Multiple Operating Systems:

- Multi-booting
- Run from USB
- Run from live disc

INTRO TO VIRTUALIZATION

Creating a virtual (rather than actual) version of something, including but not limited to a virtual computer hardware platform, operating system, storage device, or computer network resources.

Virtualization is a fundamental part of cloud computing

Hypervisor: A piece of computer software, firmware or hardware that creates and runs virtual machines.

- Type 1: Native/Bare-metal: Run directly on hardware
- Type 2: Hosted: Run on a conventional OS

ADVANTAGES OF VIRTUALIZATION

Often free or relatively low cost

Easy installation and uninstallation

Less hardware hassles

Hands-on learning of new kinds of technology

Use software normally unavailable on your machine

Test software/hardware in a safe environment

Save energy

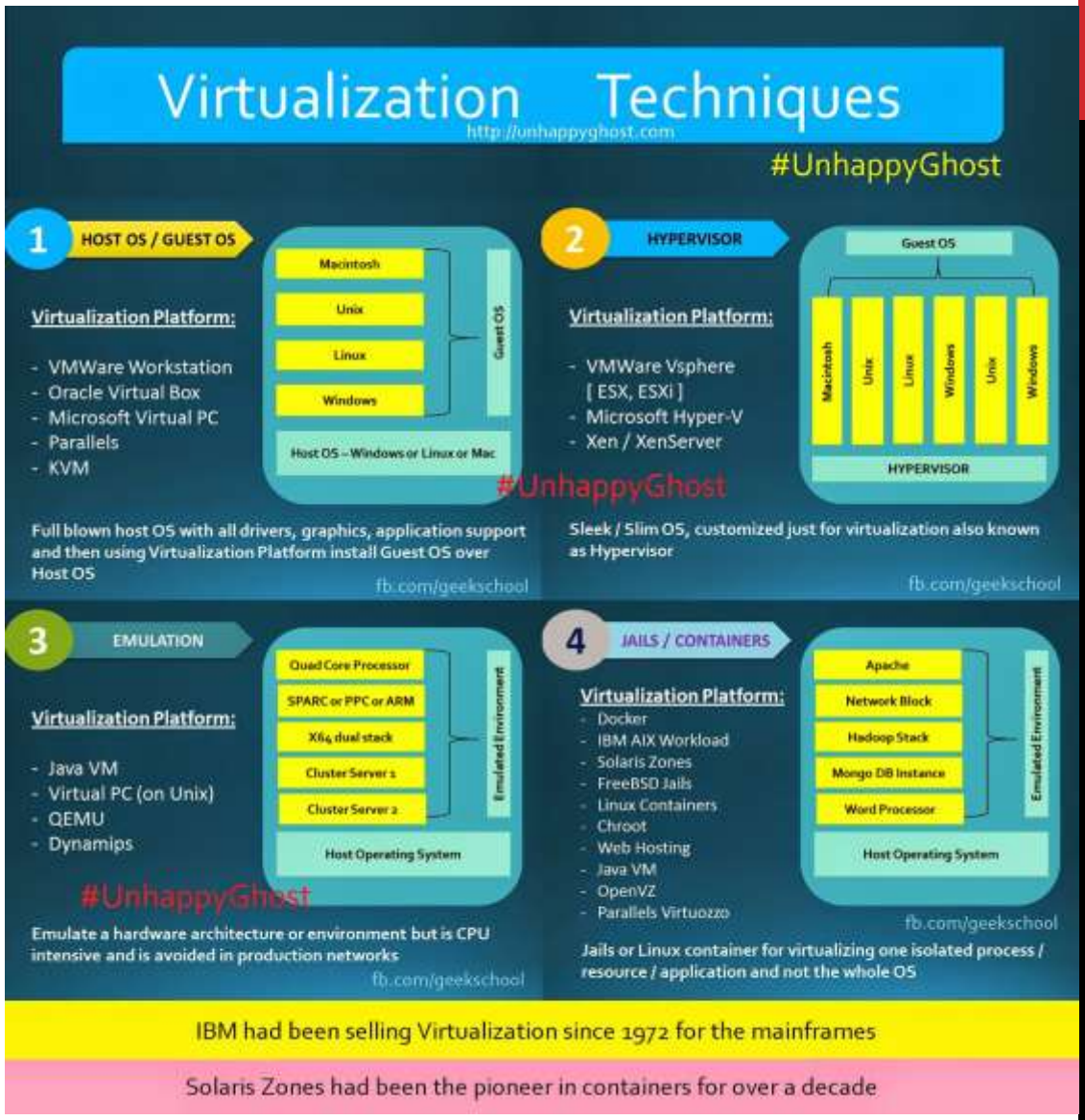
VIRTUALIZATION TECHNIQUES

Host/Guest OS: OS functions as a host, virtual machines are guests/clients. Uses Type 2 hypervisors.

Hypervisor: Uses a Type 1 hypervisor and a custom minimal OS.

Emulation/Simulation: Creates custom virtual hardware/software to imitate real hardware/software.

Jails/Linux Containers: Virtual applications run on a host OS.



COMMON USE OF DIFFERENT VIRTUALIZATION TECHNIQUES

Host/Guest OS: Virtual Desktops.

Hypervisor and Jails: Virtual Servers

Emulators: Custom/Embedded hardware environments and accompanying software (e.g. video game emulators)

We will focus on Host/Guest OS

VIRTUAL SECURITY

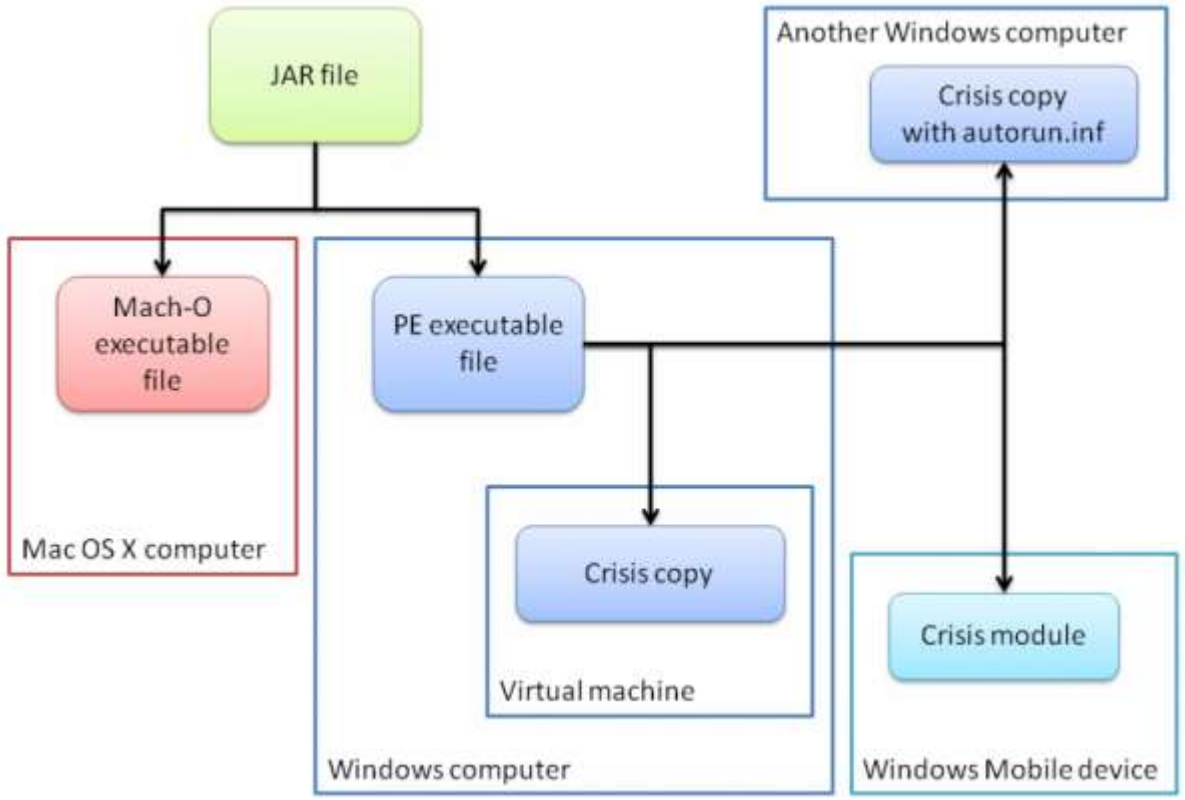
Virtual machines have some unique security considerations, as well as applications

Virtual machines can still get viruses and can transmit them to your physical machine, or vice versa

Few viruses target VMs, one such virus is detailed on the next page

VMs you plan on keeping should be protected like their physical counterparts

VM-CENTRIC VIRUS “CRISIS”



VIRTUALIZATION-CENTRIC SECURITY TECHNIQUES

Sandboxing: Run risky applications in a VM that is disconnected from your host OS and network

Server Isolation: Servers running on the same machine can be isolated virtually either by virtualizing them or by running each in its own VM

Virtual Honeypots and Honeynets: Honeypots are computers designed to lure attackers/malware/spam away from your actual computer. Honeynets are n

DEMO

http://kanishkashowto.com/2013/09/03/how-to-install-kali-linux-in-virtualbox-step-by-step-guide/

NOTES FOR DOI

Oracle Virtual Box and VMWare Player are good FREE virtualization software

You may have to download operating systems to mount (.iso files)

Some processors have virtualization technology that needs to be enabled via the BIOS menu (your software should indicate such)

Some of your hardware might not jive with your virtual machines

Some Linux and OS-X distributions default to Console view (text only) when you install them, type “startx” to switch to GUI view