

# Hacking the future with USB HID

**Nikhil “SamratAshok” Mittal**  
Hacker



Session ID:

Session Classification:

**RSACONFERENCE**  
**C H I N A 2012**  
**RSA信息安全大会2012**

# About Me

- SamratAshok
- Twitter - @nikhil\_mitt
- Blog – <http://labofapenetrationtester.blogspot.com>
- Creator of Kautilya, Mareech and Nishang
- Interested in Offensive Information Security, new attack vectors and methodologies to pwn systems.
- Previous Talks
  - Clubhack'10, Hackfest'11, Clubhack'11, Black hat Abu Dhabi'11, Black Hat Europe'12, Troopers'12, PHDays'12, Black Hat USA'12
- Upcoming Talks
  - Talk at EUsecWest'12
  - Training at GrrCON'12

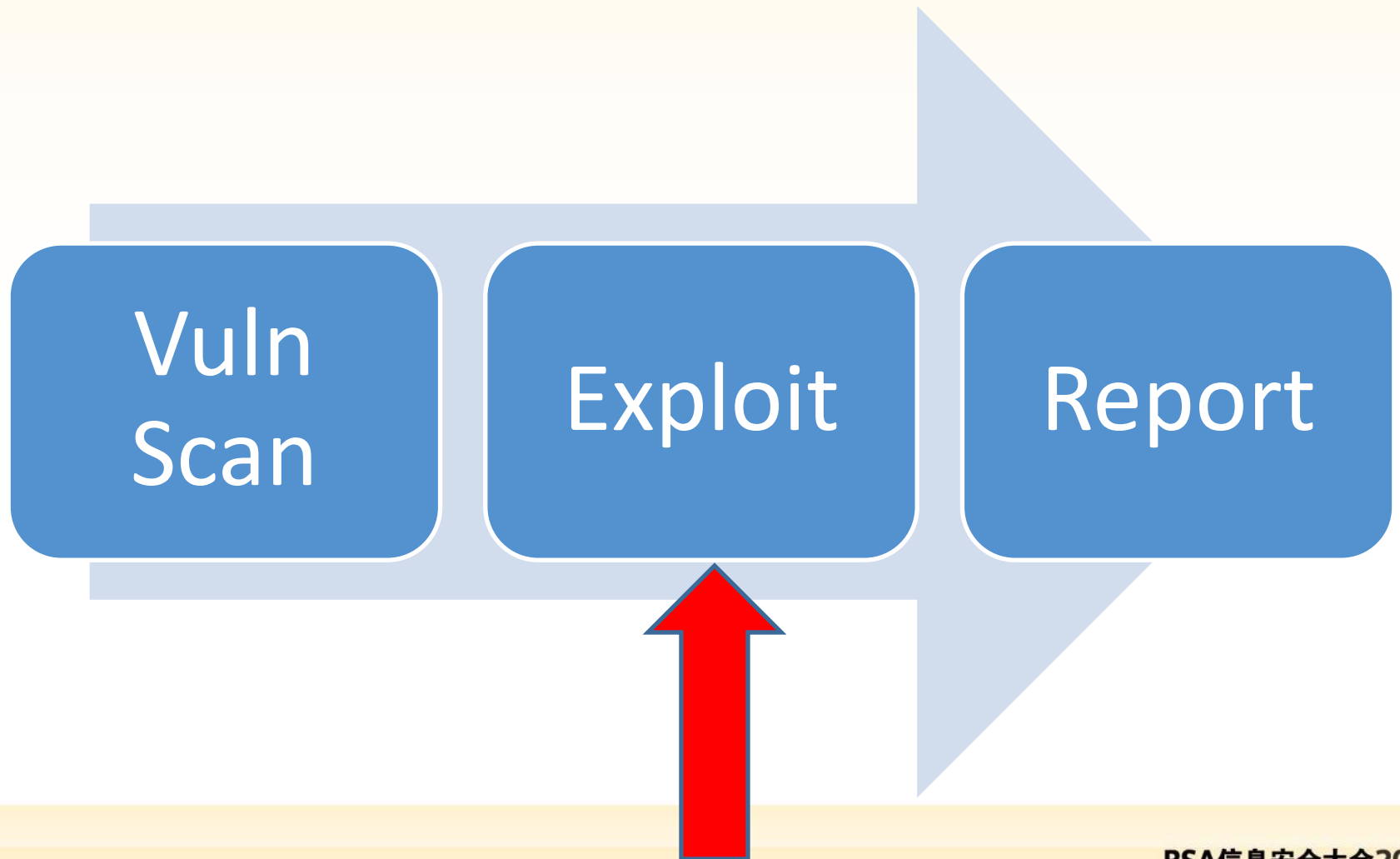
# Agenda

- Human Interface Devices
- Using HID in Penetration Tests
- HID of choice – Teensy++
- How we will use Teensy++?
- Windows Family
- Mac OS X Family
- Kautilya
- Attacks Demo (on Windows 8 and Mountain Lion )
- Comparison
- Future of Attacks
- Limitation
- Defense
- Conclusion

# A typical Pen Test Scenario

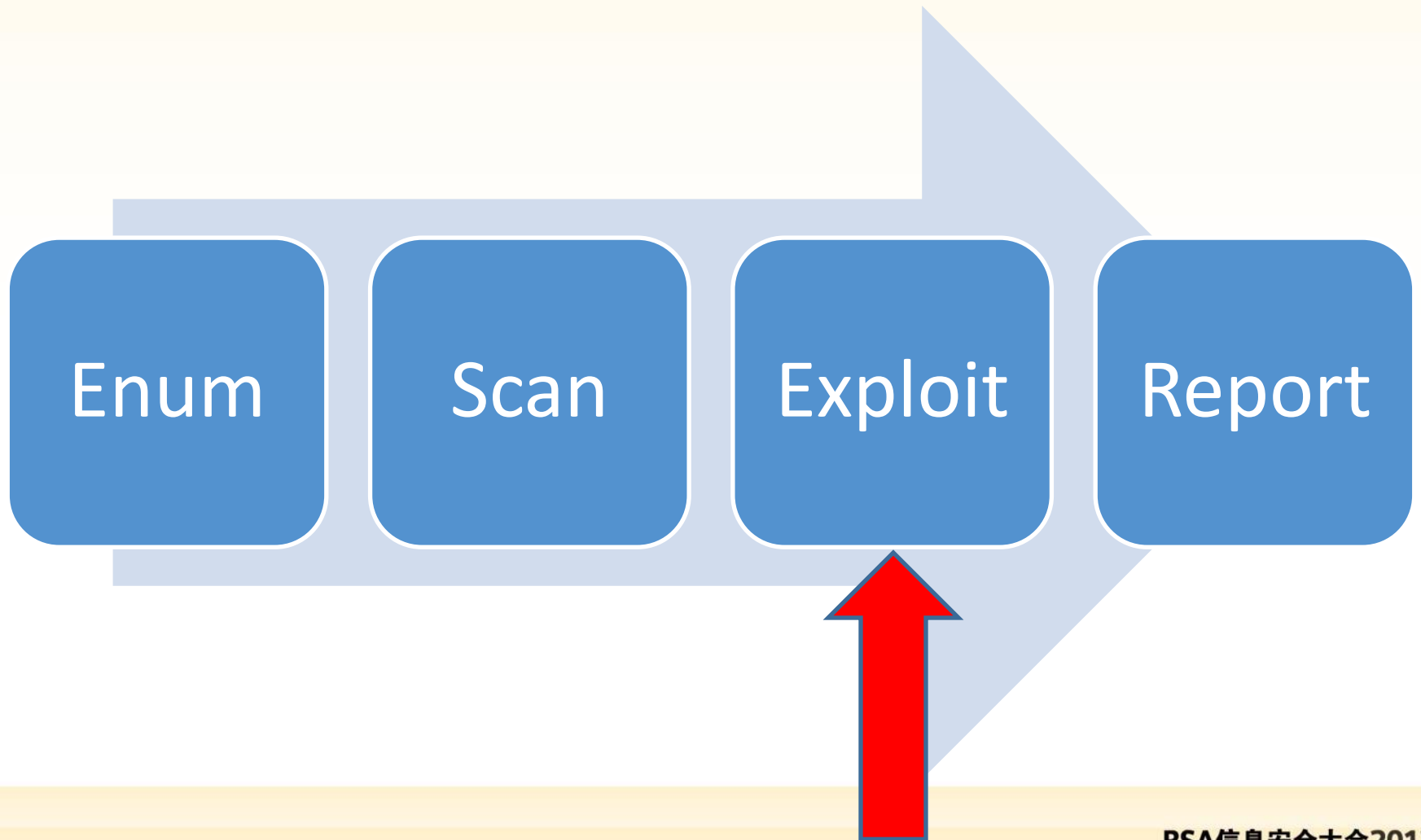
- A client engagement comes with IP addresses.
- We need to complete the assignment in very restrictive time frame.
- Pressure is on us to deliver a “good” report with some high severity findings. (That “High” return inside a red colored box)

# How the threats are Tested

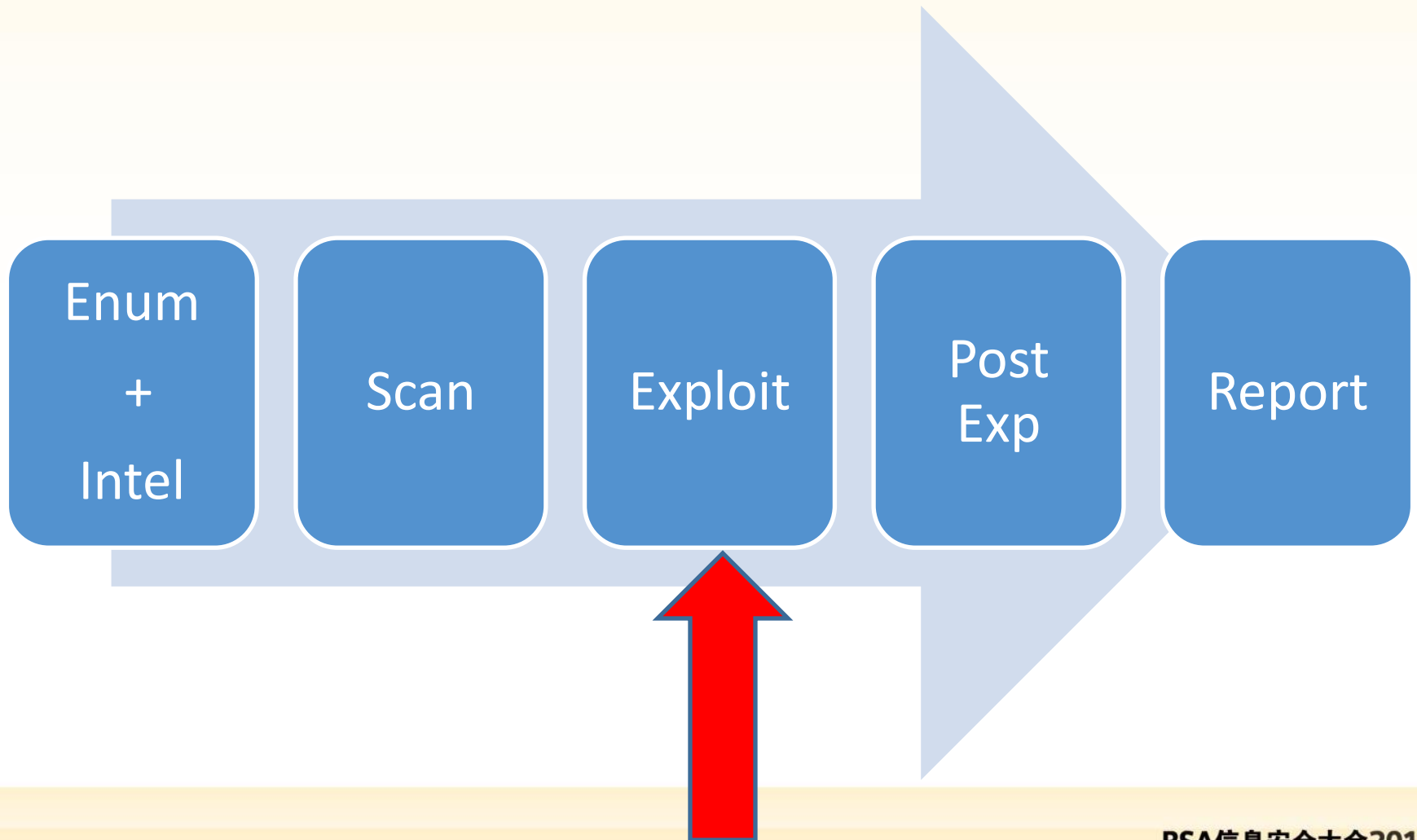


- This is a best case scenario.
- Only lucky ones find that.
- Generally legacy Enterprise Applications or Business Critical applications are not upgraded and are the first targets.
- There is almost no fun doing it that way.

# Some of us do it better



# Some of us do it even better



# Why do we need to exploit?

- To gain access to the systems.
- This shows the real threat to clients that we can actually make an impact on their business. No more “so-what” 😊
- We can create reports with “High” Severity findings which bring \$\$\$

# What do we exploit?

- Memory Corruption bugs.
  - Server side
  - Client Side
- Mis-configurations
- Open file shares.
- Sticky slips.
- Man In The Middle (many types)
- Unsecured Dumpsters
- Humans
- <Audience>

# Worse Scenario

- Many times we get some vulnerabilities but can't exploit.
  - No public exploits available.
  - Not allowed on the system.
  - Countermeasure blocking it.
  - Exploit completed but no session was generated :P

# Worst Scenario

- Hardened Systems
- Patches in place
- Countermeasures blocking scans and exploits
- Security incident monitoring and blocking
- No network access
  
- We need alternatives.

# Need for new methods to break into systems

- Breaking into systems is not as easy as done in the movies.
- Those defending the systems have become smarter and it is getting harder to break into “secured” environments.
- Everyone is breaking into systems using the older ways, you need new ways to do it better.

# Human Interface Devices

- Wikipedia – *“A human interface device or HID is a type of computer device that interacts directly with, and most often takes input from, humans and may deliver output to humans.”*
- Mice, Keyboards and Joysticks are most common HID.
- What could go wrong?

# Using HID's in Penetration Tests

- Human Interface Devices are trusted by Operating Systems.
- Countermeasures like Anti Virus do not care for such devices.
- The way we use it, using HID for offensive security is equivalent to sitting in front of the target system as a user.
- The attack scenarios are large in number and may have severe impact.

# HID of choice - Teensy++

- A USB Micro-controller device from pjrc.com
- Storage of about 130 KB.
- We will use Teensy ++ which is an updated version of Teensy.
- A cheap device, costs only \$24.
- It uses an Atmel based processor.



# HID of choice - Teensy++

- It could be used as Keyboard/Mouse/Joystick.
- The device is easily programmable using C or C++ type syntax using Arduino Development Environment with Teensyduino plugin.
- The device works with many Operating Systems.
- It is small in size.

# How we will use Teensy++?

- As a programmable keyboard.
- We will program the device to do a defined set of activities when it is connected to a system.
- We will utilise the privileges of the currently logged in user and any higher privileges accessible to the user.
- Aim is to mimic a user sitting in front of the target.

# Windows Family

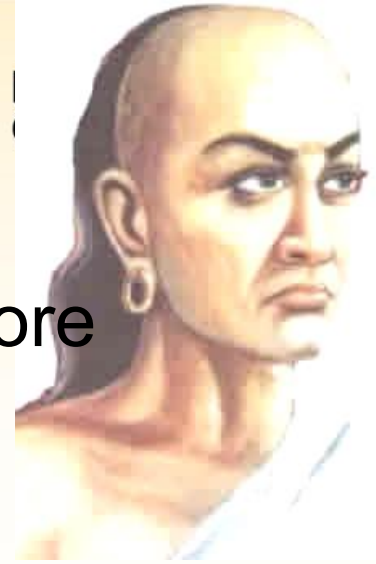
- A user is notified when a new device is connected.
- It takes 20-25 seconds while the driver for the device gets loaded.
- A device can type really fast on Windows machine thanks to large USB keyboard buffer of Windows.
- If PowerShell is used some really powerful things could be done.

# OS X Family

- A user is not notified if a USB device is connected.
- It takes 10-15 seconds while the device is detected and loaded.
- The device cannot type very fast.
- Built-in scripting languages make payloads powerful.

# Kautilya

- It is a toolkit which aims to make HID more useful in Penetration Tests.
- Named after Chanakya a.k.a. Kautilya.
- Written in Ruby.
- It's a menu drive program which let users select and customize payloads.
- Aims to make HID part of every Penetration tester's tool chest.
- Contains payloads for Windows, Linux and OS X.



# Payloads in Kautilya

- Payloads are tested on Teensy without SD Card.
- Pastebin is extensively used for uploads and downloads.
- Payloads are commands, powershell scripts or combination of both.
- Payload execution depends on privilege of user logged in when the device is plugged in.

# Attacks Demo (on Windows 8 and Mountain Lion )

- Let us have a look at three attacks on both
  - Download and execute shellcode.
  - Reverse shell using built-in features.
  - DNS TXT Code Execution.

# Comparison

Attribute	Windows 8	Mac OS X Mountain Lion
Detection or blocking of USB HIDs	Shows a balloon. Easy to prevent installation of removable devices using Group policies.	No information to user. Not easy to block a device.
Response to a very fast keyboard input	Possible to send input really fast.	Delays must be introduced between the keyboard inputs.
Trust on end user (as we are simulating one)	For sensitive functions a UAC prompt is shown.	Sudo is required for sensitive functions.

# Pen Test Stories

## Library Fun

- Internal PT for a large media house.
- The access to network was quite restrictive.
- The desktops at Library were left unattended many times.
- Teensy was plugged into one system with a sethc and utilman backdoor.
- Later in the evening the system was accessed and pwnage ensued.

# Pen Test Stories

## Breaking the perimeter

- A telecom company.
- A perimeter check for the firm was to be done.
- The Wireless rogue AP payload was used and Teensy was sold to the clients employees during lunch hours.
- Within couple of hours, we got a wireless network ready with an administrative user and a bind shell.

# Pen Test Stories

## Help by the Helpdesk

- A pharma company.
- A user's data card was replaced with a Teensy inside the data card's cover.
- The payload selected was Keylogger.
- "Data card" obviously didn't work and multiple keyloggers were installed, for the user and the helpdesk.
- Helpdesk guys had access to almost everything in the environment and over a workday, it was over.

# Limitations with Teensy

- Limited storage in Teensy. Resolved if you attach a SD card with Teensy.
- Inability to “read” from the system. You have to assume the responses of victim OS and there is only one way traffic.
- Inability to clear itself after a single run.

# Limitations with Kautilya

- Many payloads need Administrative privilege.
- Lots of traffic to and from pastebin.
- Some times payloads are not stable.
- For payloads which use executables you manually need to convert and paste them to pastebin.

# Future of Attacks

- Improvement in current payloads.
- Use some payloads as libraries so that they can be reused.
- More payloads for Non-Windows platform.
- Implementation of newer payloads.
- Reliable user activity detection.

# Defence

- For Windows systems, use Group Policy to “Prevent Installation of Removable Devices”.
- For Mac OS X, udev rules may be used.
- Best defence is to physically block USB ports or lock the existing devices to the ports.

# Conclusion

- USB HID attacks are real threats and here to stay.
- This is because Operating System trust itself and its users.
- Security ends with trust.

# Thank You

- Questions?
- Insults?
- Feedback?
  
- Kautilya is available at <http://code.google.com/p/kautilya/>
- Follow me @nikhil\_mitt
- <http://labofapenetrationtester.blogspot.com/>

