Hacking Google ChromeOS

Matt Johansen
Team Lead
matt.johansen@whitehatsec.com
@mattjay

Kyle Osborn
Application Security Specialist
kyle.osborn@whitehatsec.com
@theKos

August 2011

special thanks to:
Google's Security Team
Jeremiah Grossman
Chris Evans
Sumit Gwalani

WhiteHat SECURITY

Who are we?

Kyle & Matt are both part of the Threat Research Center at WhiteHat Security and manually assess a large portion of WhiteHat's 4,000+ websites.

- Matt:
  - *Application Security Engineer turned Team Lead.*
  - *Background in Penetration Testing as a Consultant.*
  - *Bachelor of Science in Computer Science from Adelphi University*
- Kyle:
  - *Application Security Specialist*
  - *Primary focus on Offensive Security Research*
  - *Likes to push the Big Red Button*

WhiteHat Security Company Overview

- WhiteHat Security: end-to-end solutions for Web security
- WhiteHat Sentinel: SaaS website vulnerability management
  Combination of cloud technology platform and leading security experts turn security data into actionable insights for customers
- Founded in 2001; Sentinel Premium Edition Service launched in 2003
- 400+ enterprise customers, 4,000 sites under management
- Most trusted brand in website security

ChannelWeb 20 Coolest Cloud Security Vendors

Deloitte. 2008 Technology Fast 50 Silicon Valley

Social Security Blogger Awards 2010 at BlogU Security Bloggers Meet-Up @ RSA Conference

Red Herring N. America Winner 100

OnDemand 2010 Top 100 Private Companies Competition

Inc. 500

2007 InfoWorld CTO 25

BANK TECHNOLOGY NEWS btn The FutureNow List

WhiteHat SECURITY

Google Cr-48 Beta Laptop

- First Chrome OS dedicated device
- Application to be a Beta Tester open to public
- WhiteHat one of few security companies to test it first

Chrome OS

*"**The time for a Web OS is now**"* – Eric Schmidt

What we know:

- Revolves around the browser
- Virtually nothing stored locally
- Cloud heavy (re: reliant)
- *Fast!*

Google Chrome OS

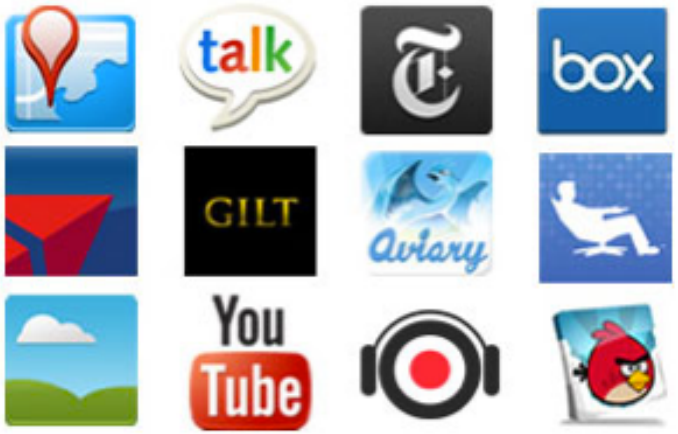
WhiteHat SECURITY

Chrome OS (cont'd)

Nothing stored locally = no usual software suspects.

Mobile = App Crazy

Chrome OS = Extension Crazy



In order to get usability / functionality out of a locked up device user's must use what is available.

WhiteHat SECURITY

What Does A Hacker See?

New attack surface!

With all of these new extensions that aren't necessarily developed by Google or any reputable company, security vulnerabilities are bound to be plentiful.

Let the Hacking Begin!

ScratchPad

Preinstalled note-taking extension

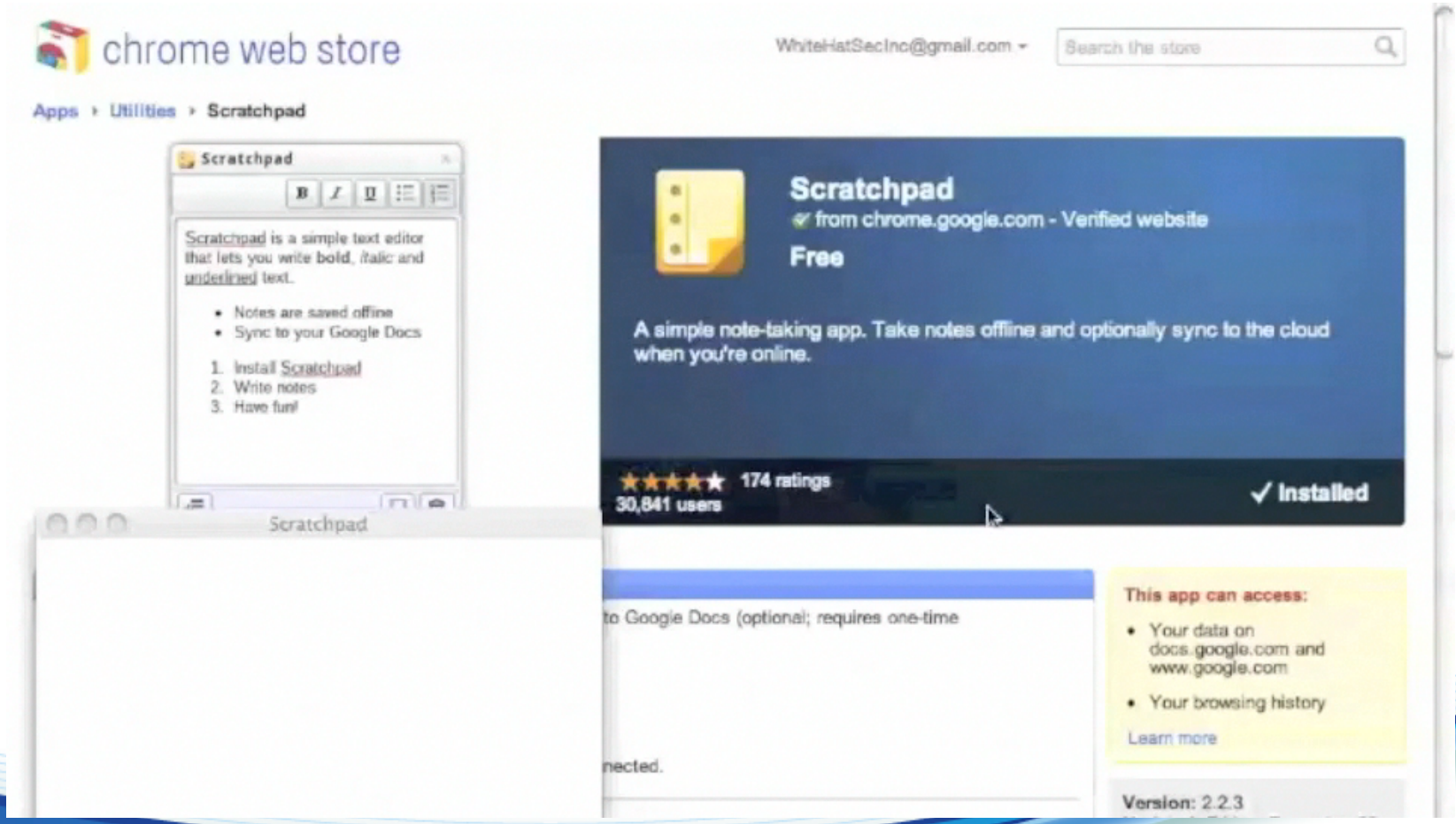
Auto Sync feature to Google Docs "ScratchPad" Folder

Google Docs "Feature" – Folder/Doc sharing. No permission needed!

WhiteHat SECURITY

ScratchPad Video demo

Google fixed Scratchpad XSS very quickly but we have a video demo.



Permission Structure

Why are Extensions any different?

- Individual extensions have unique permissions
- Use chrome.* API
- **Permissions are set by the 3rd party developer**
- Some extensions require permission to talk to every website
- Similar to Mobile Apps

Overview of chrome.* APIs

- chrome.bookmarks
- chrome.cookies
- chrome.history
- chrome.windows
- chrome.tabs

- URL match pattern = domains an extension **can** access
  - ***://*/*** <- Do Not

```
16    },
17    "permissions": [
18      "unlimited_storage",
19      "notifications",
20      "chrome://favicon/",
21      "tabs"
22    ]
23  }
```


Page 11
© 2011 WhiteHat Security, Inc.

Usual suspects

What we are looking for: (What you are watching out for)

- RSS Readers
- Mail notifiers
- Note takers
- Anything that takes input from somewhere and displays it to the user

© 2011 WhiteHat Security, Inc.

Easy to exploit

- Why worry about native code execution?
- XSS gives hackers everything we could ask for and more.
- Exploit development is hard. Javascript is easy.

"Javascript so easy I could do it"
- Small Child

© 2011 WhiteHat Security, Inc.

Malicious Extension Demo

What can we do with a very vulnerable extension with wide open permissions which do exist in the wild.

WhiteHat SECURITY

Owning the un-ownable

Example: LastPass.com(LP):

- No vulnerability. (Fixed the one I found immediately)
- Find a vulnerability in Joe-Schmoe RSS reader.
- Discover LastPass.com in bookmarks/history, plus LP extension installed.
- Spawn a new window with LastPass.com
- Auto-logged in because of LP extension feature
- Inject code to steal your local crypto key & LP DB.
- Decrypt on my side with key & DB
- Profit

Page 15
© 2011 WhiteHat Security, Inc.

Let's make it easy. I mean really easy.

- How can we simplify this even more?

- Reuse attacks!
- Incomes BeEF - Browser Exploitation Framework
  1. Discover vulnerable extension
  2. Inject BeEF hook.js
  3. Utilize BeEF to maintain access
  4. Replay commands as necessary

Page 16
© 2011 WhiteHat Security, Inc.

Let's make it easier. I mean really easy. (cont'd)

- BeEF plugins released!

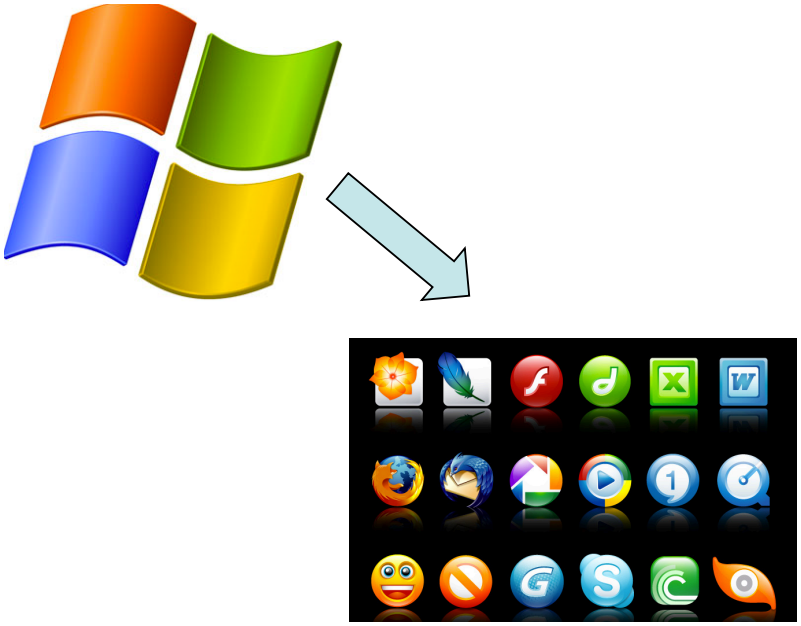
- **check_permissions**
  - Check extension permissions
- **execute_tabs**
  - Execute Javascript on all tabs
- **inject_beef**
  - Inject BeEF into all tabs open
- **grab_google_contacts**
  - Download Google Contacts
- **port_scan**
  - Scan ports on local network using XHR
- **send_gvoice_sms**
  - Send text messages via logged in Google Voice

Page 17
© 2011 WhiteHat Security, Inc.

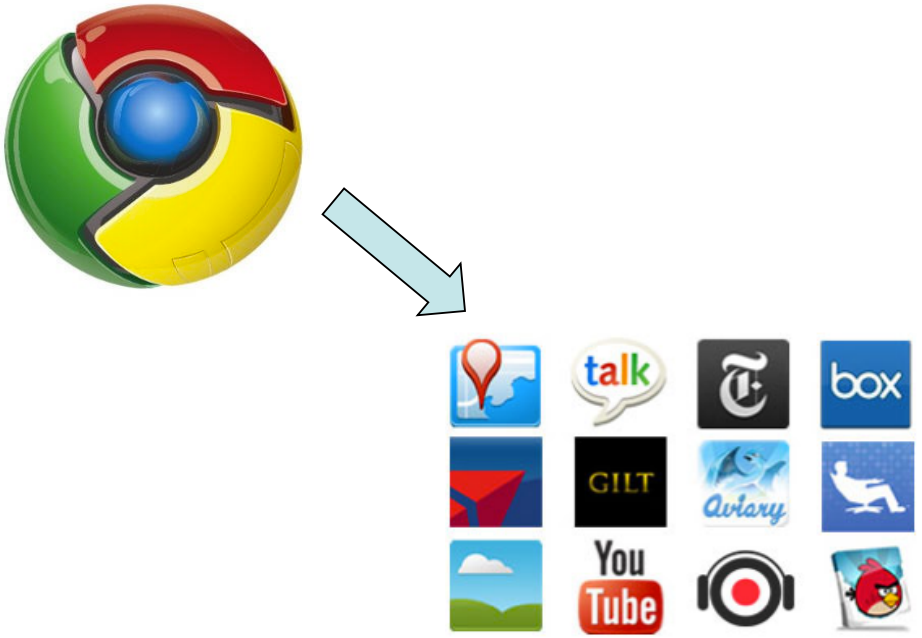
Browser -> Extension Trust Model

Taking the old Software Security Model and moving it to the cloud.

Software Security Model



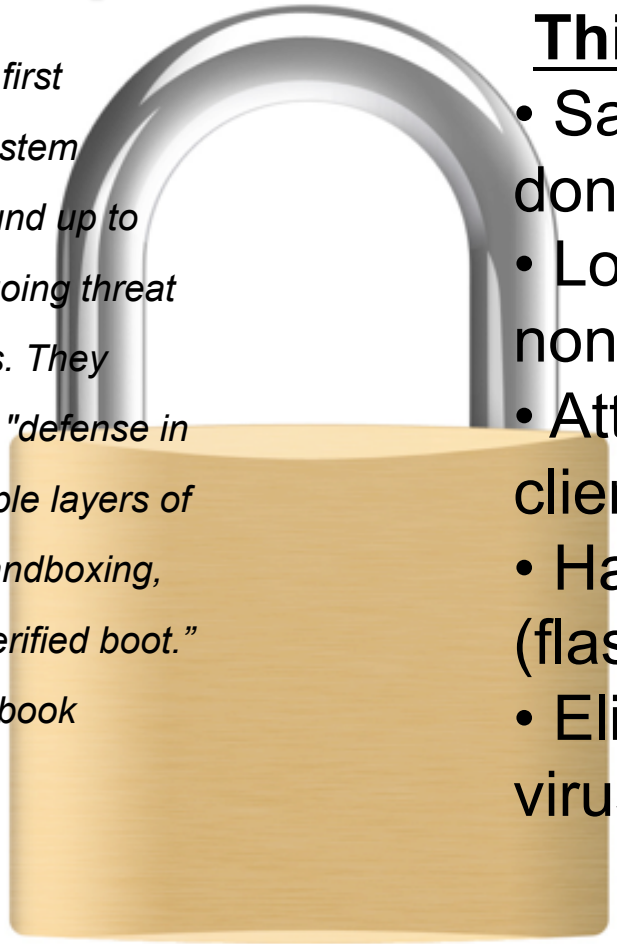
Browser Extension Trust Model



WhiteHat SECURITY

Security Implications

"Chromebooks run the first consumer operating system designed from the ground up to defend against the ongoing threat of malware and viruses. They employ the principle of "defense in depth" to provide multiple layers of protection, including sandboxing, data encryption, and verified boot."
– Google.com/Chromebook



Things Done Very Well

- Sandboxing tabs so they don't talk to each other
- Local storage is virtually non existent
- Attack surface limited to client side browser exploits
- Handles own plugins (flash, pdfs, etc.)
- Eliminates most modern virus / malware threats

Please Remember to Complete Your Feedback Form

Thank You!

Q&A?

Matt Johansen
Team Lead
matt.johansen@whitehatsec.com
@mattjay

Kyle Osborn
Application Security Specialist
kyle.osborn@whitehatsec.com
@theKos

special thanks to:
Google's Security Team
Jeremiah Grossman
Chris Evans
Sumit Gwalani

WhiteHat SECURITY