Disrupting the Malware Lifecycle

Louis F. DeKoven, ldekoven@cs.ucsd.edu

Ph.D. Candidate, UC San Diego

Naval Postgraduate School, Tuesday, October 24th, 2017

Motivation

- Malware is everywhere
  - Browsing the Internet
  - Using social media services

Motivation

- Malware is everywhere
  - Browsing the Internet
  - Using social media

- There is no silver bullet
  - No one-fix-all solution to malware

Motivation

- Malware is everywhere
  - Browsing the Internet
  - Using social media
- There is no silver bullet
  - No one-fix-all solution
- Explore new techniques
  - Disrupt malware across lifecycle

Disrupting the Malware Lifecycle

- Infected Devices
  - Detecting and cleaning up new malware infections

Disrupting the Malware Lifecycle

- Infected Devices
  - Detecting and cleaning up new malware infections

- User Behaviors and Security Outcomes
  - Putting security outcomes on an empirical basis

Malicious Browser Extensions at Scale

Bridging the Observability Gap between Web Site and Browser

Louis F. DeKoven[1], Stefan Savage[1], Geoffery M. Voelker[1], Nektarios Leontiadis[2]

[1]UC San Diego, [2]Facebook

Attacks on Social Media

- Social media is targeted by malware
  - Reach a large number of users quickly
  - Users inherently trust content within a social network

Attacks on Social Media

- Social media is targeted by malware
  - Reach a large number of users quickly
  - Users inherently trust content within a social network
- Malware infects user's browser then
  - Infect other social media users
  - Steal the user's passwords

Attacks on Social Media

- Social media is targeted by malware
  - Reach a large number of users quickly
  - Users inherently trust content within a social network
- Malware infects user's browser then
  - Infect other social media users
  - Steal the user's passwords
- Leverage the vantage point of a social network to
  - Detect devices infected with malware
  - Clean up malware from infected devices

Objectives

- Detect and label malicious browser extensions quickly
  - Google Chrome
  - Mozilla Firefox
- Automatically cleanup infected devices
- Detect new malicious browser extensions automatically

Objectives

- Detect and label malicious browser extensions quickly
  - Google Chrome
  - Mozilla Firefox
- Automatically cleanup infected devices
- Detect new malicious browser extensions automatically

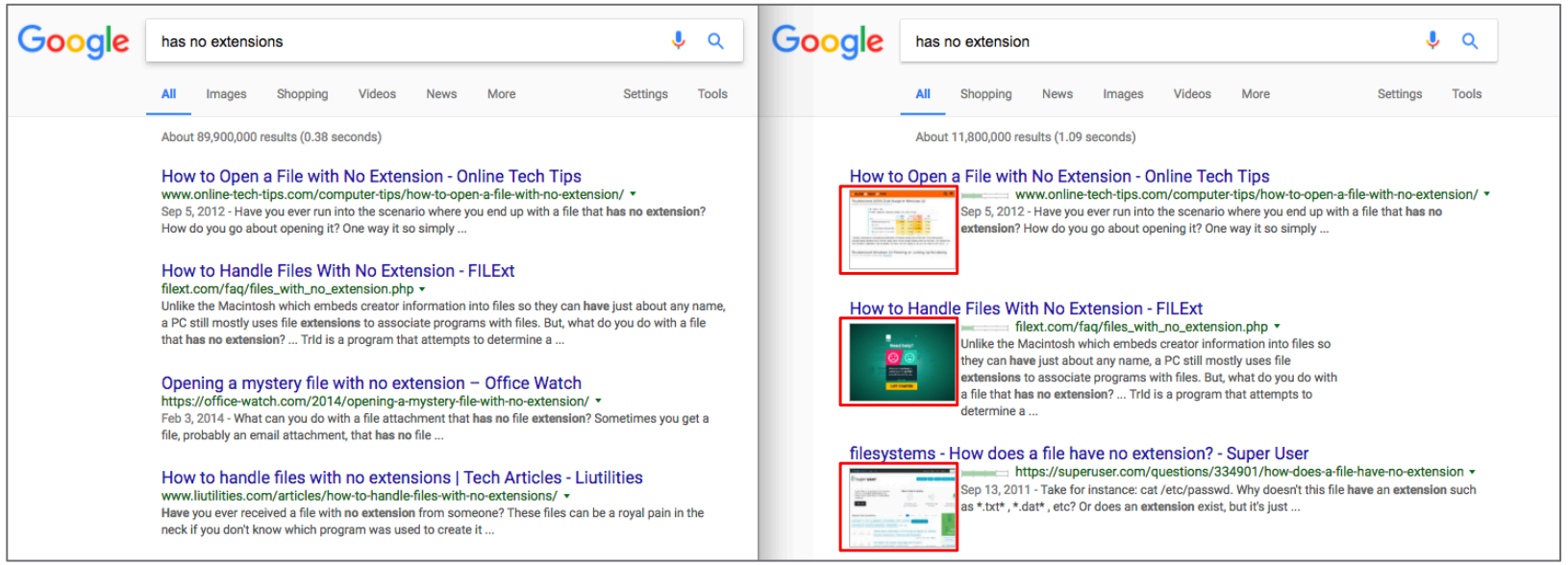
Malicious Browser Extensions (MBE): extensions that take actions on behalf of a user without their consent, or replace Facebook's key functionality or content.

Browser Extensions 101

- Enhance user experience beyond a Web page
- Can change visual appearance of Web pages
- Can change how the browser interacts with Web pages

Browser Extensions 101

- Enhance user experience beyond a Web page
- Can change visual appearance of Web pages
- Can change how the browser interacts with Web pages



No extension Extension

Browser Extensions 101

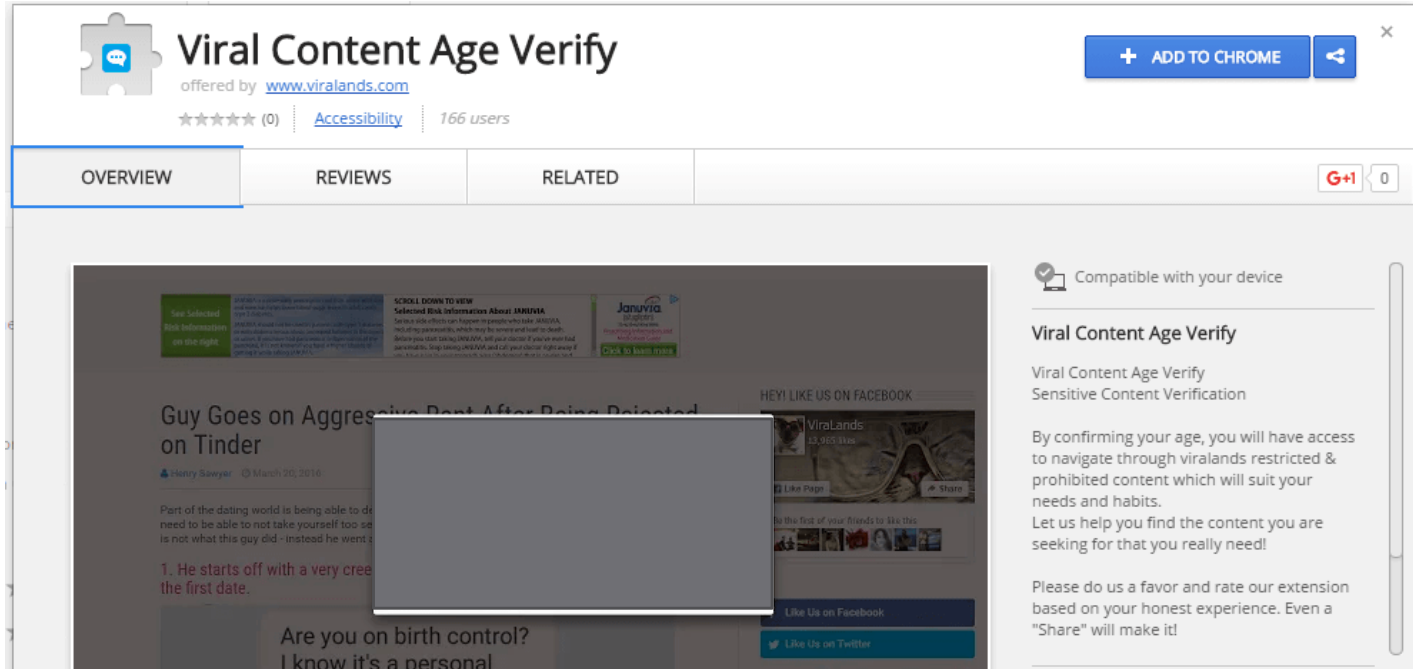
- Enhance user experience beyond a Web page
- Can change visual appearance of Web pages
- Can change how the browser interacts with Web pages
- How?
  - Have elevated set of privileges

Browser Extensions 101

- Enhance user experience beyond a Web page
- Can change visual appearance of Web pages
- Can change how the browser interacts with Web pages
- How?
  - Have elevated set of privileges
    - Modify HTTP headers
    - Change Content Security Policy
    - Rewrite any Web site content

Browser Extensions 101

- Example MBE targeting Facebook
  - Steals user's Facebook access token
  - Generates likes
  - Subscribes to YouTube channels
  - And more…



https://kjaer.io/extension-malware/

Extension Manifest File

- Metadata file containing information about the extension
  - Name/Description
  - Permissions
  - Scripts

https://kjaer.io/extension-malware/

Extension Manifest File

- Metadata file containing information about the extension
  - Name/Description
  - Permissions
  - Scripts

- MBE Permissions
  - all_urls: “Read and modify all your data on all websites you visit”
  - tabs: “Access your browsing activity”

https://kjaer.io/extension-malware/

Extension Manifest File

- Metadata file containing information about the extension
  - Name/Description
  - Permissions
  - Scripts

- MBE Permissions
  - all_urls: “Read and modify all your data on all websites you visit”
  - tabs: “Access your browsing activity”
- MBE Scripts
  - Specifies scripts that run **persistently**
  - Persistent: can not be paused

https://kjaer.io/extension-malware/

Extension Scripts

- `background.js`
  - Should perform age verification

```
document.querySelector('#submit').addEventListener('click', function() {
        document.querySelector('#box').hidden = true;
        document.querySelector('#loading').hidden = false;

        setTimeout(function() {
                document.querySelector('#loading').hidden = true;
                document.querySelector('#done').hidden = false;
        }, (randomIntFromInterval(0, 2) / 2 + 0.5) * 1000);
});
```

https://kjaer.io/extension-malware/

Extension Scripts

- `background.js`
  - Should perform age verification
  - Instead: spins, and then displays “done”

- `query-string.js`
  - Appears to be a benign copy of a common NPM library
  - Until line **133…**

https://kjaer.io/extension-malware/

Extension Scripts

- `background.js`
  - Should perform age verification
  - Instead: spins, and then displays “done”

- `query-string.js`
  - Appears to be a benign copy of a common NPM library
  - Until line **133…**

```
var programUrl = 'http://104.131.35.136:9999/jsnew.php?id=22';
```

https://kjaer.io/extension-malware/

Extension 2nd Stage Payload

- Extension fetches code at `programURL` via `XMLHttpRequest`
- Executes response from Web request
- Gets instructions from C&C server capable of:
  - Stealing Facebook access tokens
  - Liking Facebook pages
  - Subscribing to YouTube channels
  - And more..

At the time of detection over 132,000 installs

https://kjaer.io/extension-malware/

Defending Against MBE

- Harden the browser [1,2,3]
- Detecting extensions vulnerable to Web page JavaScript[4]
- Vetting code within extension marketplaces [5]
- Dynamic analysis and sandboxing [6,7]

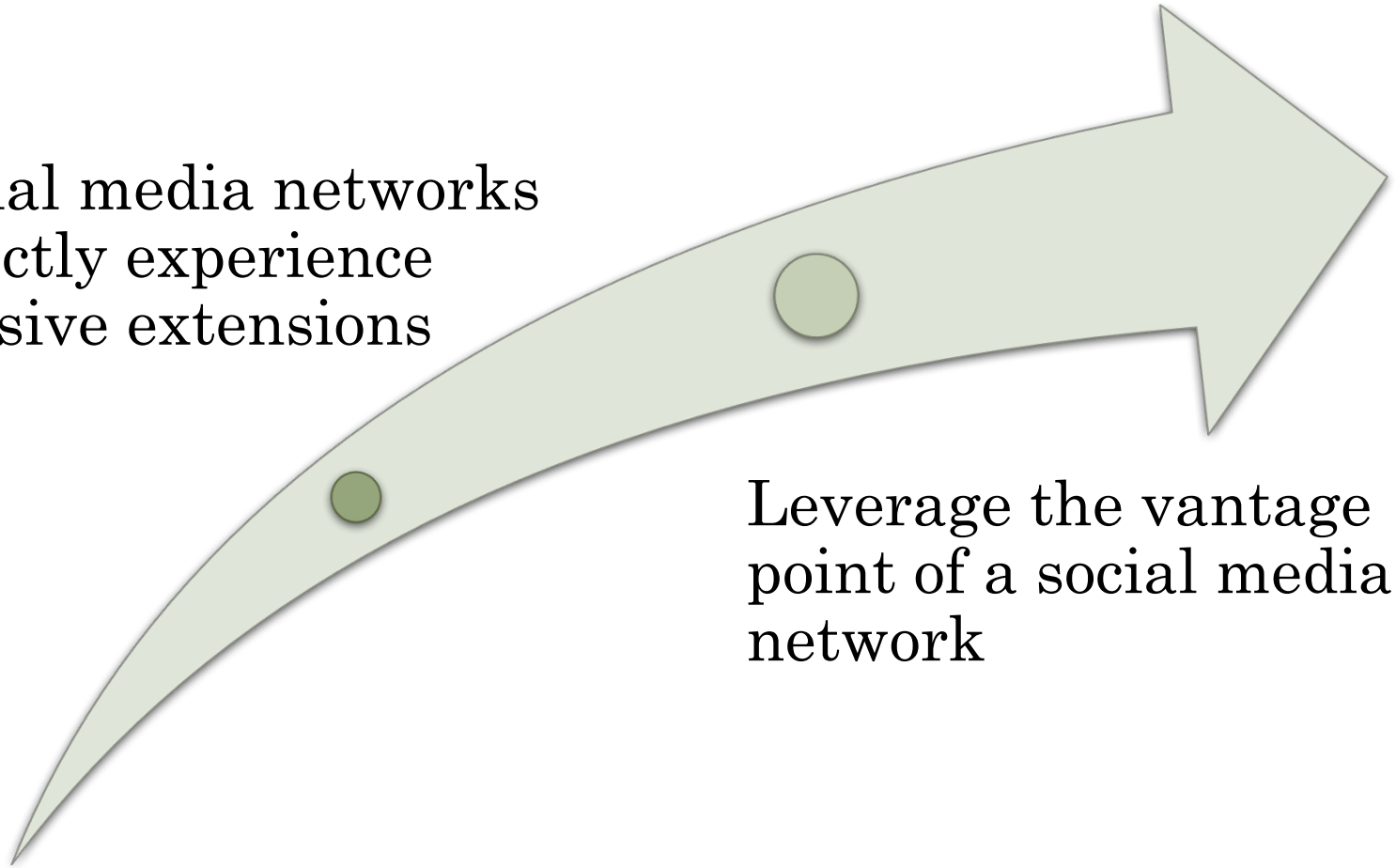
[1] V. Djeric and A. Goel. Securing Script-Based Extensibility in Web Browsers. In *Proc. of USENIX Security*, 2010.
[2] A. Guha, M. Fredrikson, B. Livshits, and N. Swamy. Verified Security for Browser Extensions. In *Proc. of IEEE S&P*, 2011.
[3] L. Liu, X. Zhang, G. Yan, and S. Chen. Chrome Extensions: Threat Analysis and Countermeasures. In *Proc. of NDSS*, 2012.
[4] M. T. Louw, J. S. Lim, and V. N. Venkatakrishnan. Enhancing web browser security against malware extensions. *Journal in Computer Virology*, 2008.
[5] H. Shahriar, K. Weldemariam, T. Lutellier, and M. Zulkernine. A Model-Based Detection of Vulnerable and Malicious Browser Extensions. In *Proc. of SERE*, 2013.
[5] S. Bandhakavi, S. T. King, M. Parthasarathy, and M. Winslett. Vetting Browser Extensions for Security Vulnerabilities with VEX. In *Proc. of USENIX Security*, 2010.
[6] A. Kapravelos, C. Grier, N. Chachra, C. Kruegel, G. Vigna, and V. Paxson. Hulk: Eliciting Malicious Behavior in Browser Extensions. In *Proc. of USENIX Security*, 2014.
[7] N. Jagpal, E. Dingle, J. Gravel, P. Mavrommatis, N. Provos, M. A. Rajab, and K. Thomas. Trends and Lessons from Three Years Fighting Malicious Extensions. In *Proc. of USENIX Security*, 2015.

It's Hard to Detect MBE

- Anti-malware products
  - May run static analysis on extension JavaScript
  - Struggle with dynamic resources
- Extension marketplaces/Browser vendors
  - May track how extensions use the browser
  - Struggle with temporal badness
- Researchers
  - May run sandboxed analysis
  - Struggle with scale and temporal badness

A Different Perspective

Social media networks directly experience abusive extensions



Leverage the vantage point of a social media network

Challenges in Detecting MBE

- How do we know what extensions are bad?
  - Facebook has to build signatures to detect MBE

Challenges in Detecting MBE

- How do we know what extensions are bad?
  - Facebook has to build signatures to detect MBE
- Facebook does not know what extensions are installed
  - Can detect user accounts acting in abusive ways

Challenges in Detecting MBE

- How do we know what extensions are bad?
  - Facebook has to build signatures to detect MBE
- Facebook does not know what extensions are installed
  - Can detect user accounts acting in abusive ways
- Facebook can not collect extensions from facebook.com due to browser security
  - Can build a binary to collect installed extensions

Challenges in Detecting MBE

- How do we know what extensions are bad?
  - Facebook has to build signatures to detect MBE
- Facebook does not know what extensions are installed
  - Can detect user accounts acting in abusive ways
- Facebook can not collect extensions from facebook.com due to browser security
  - Can build a binary to collect installed extensions

- Insight: We can link extension content to abusive content

System Methodology

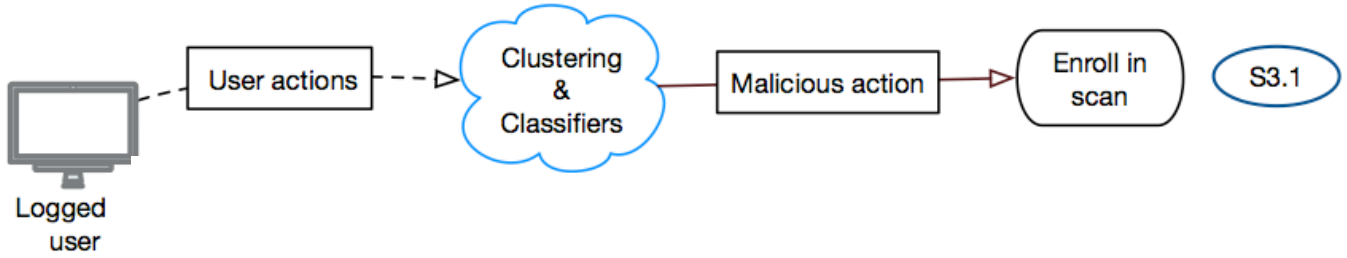
Using signals from malware within Facebook enables the detection and remove MBE at a large scale

We do this by:

- Identifying compromised Facebook accounts
- With user consent, we fetch the installed extensions from devices exhibiting malicious behavior
- Determine if the extension is malicious or benign by comparing it to abusive content (while fetching extensions)
- If the extension is malicious remove it from the user's device

System Design

- Detecting compromised user accounts



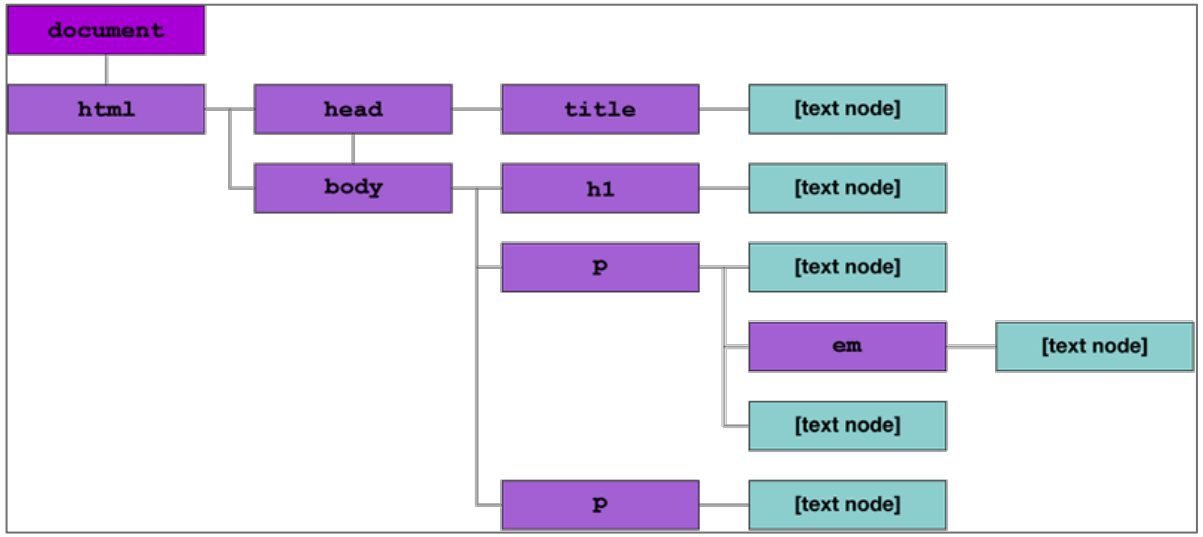
Detecting Compromised User Accounts

- Spiking content
  - Monitor time series of user activity

Detecting Compromised User Accounts

- Spiking content
  - Monitor time series of user activity

- Document Object Model (DOM) based detection
  - Periodically scan Facebook's DOM for third-party elements



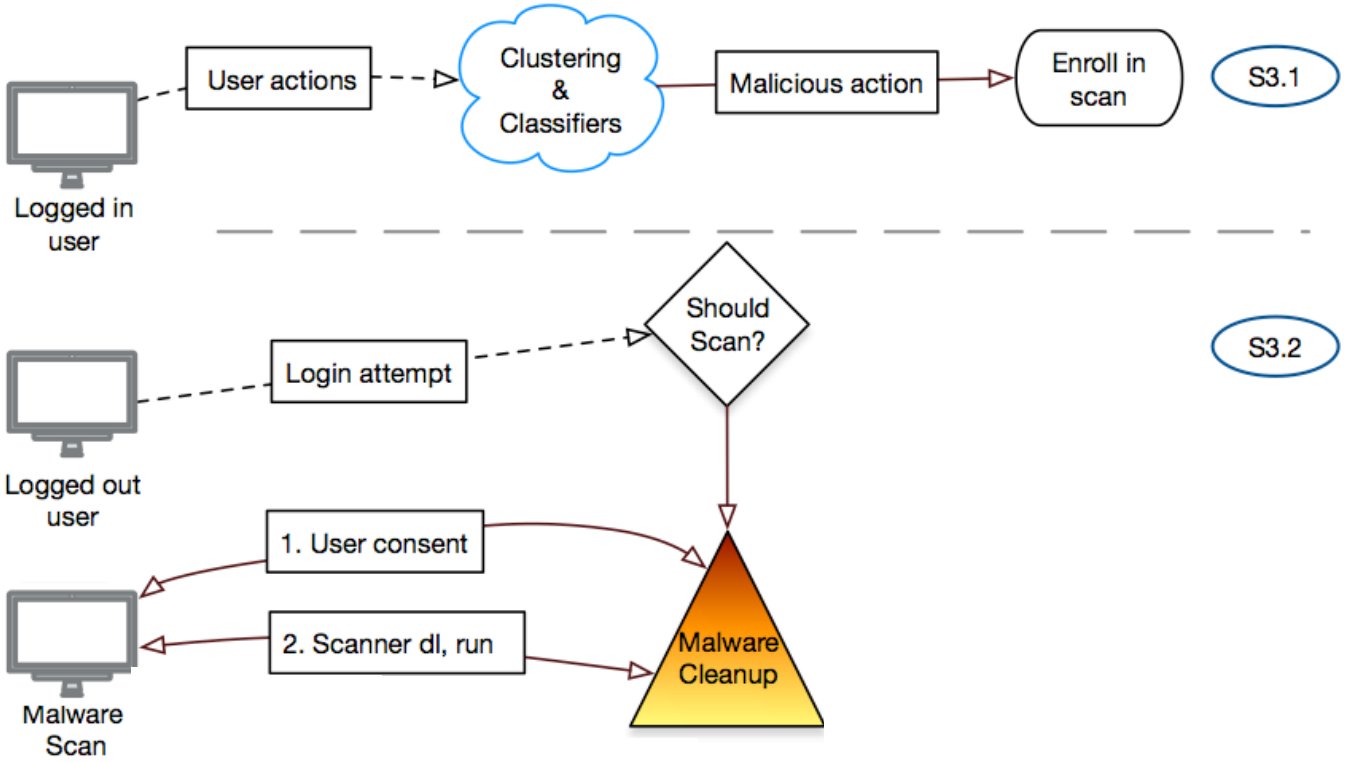
Example DOM

Detecting Compromised User Accounts

- Spiking content
  - Monitor time series of user activity
- Document Object Model (DOM) based detection
  - Periodically scan Facebook's DOM for third-party elements
- Negative feedback
  - Feedback on posted content

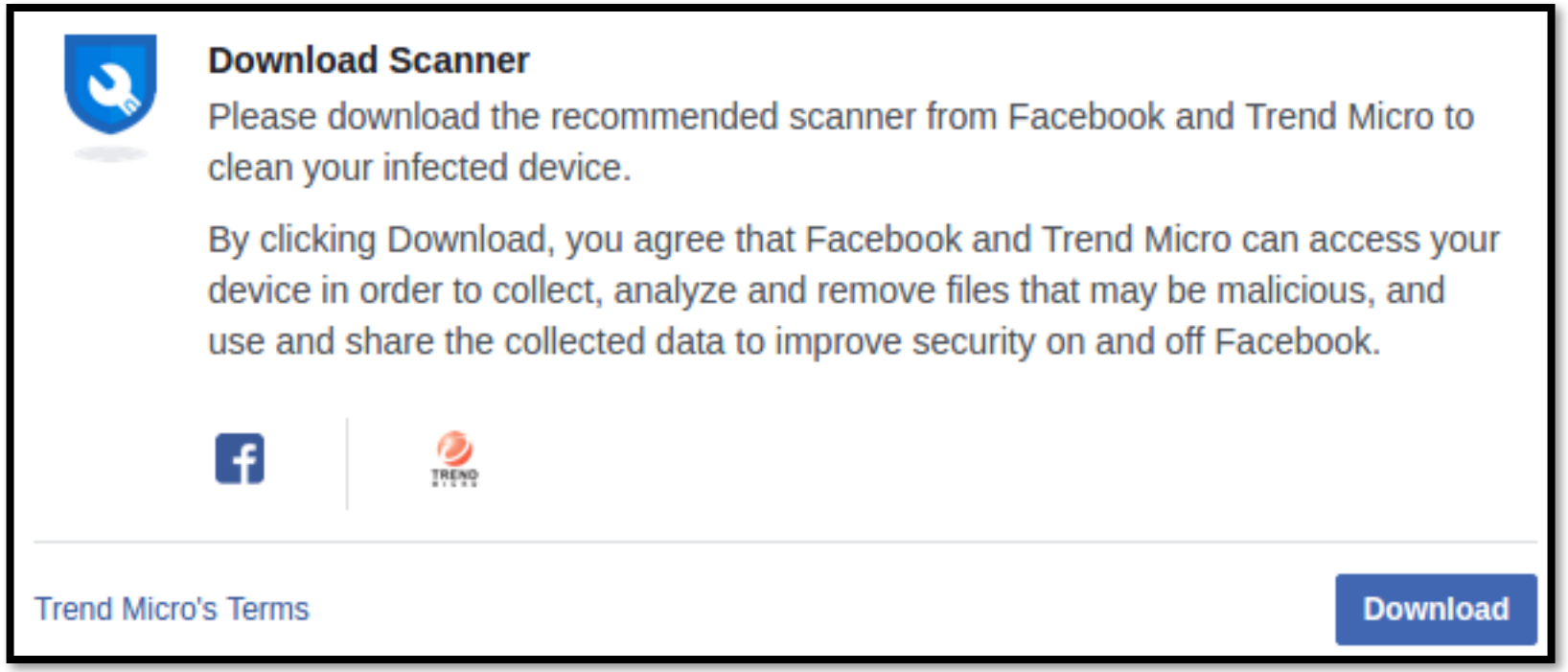
System Design

- Detecting compromised user accounts
- Anti-malware scanner



Anti-Malware Scanner

- Facebook's custom scanner is executed on the compromised device following user consent



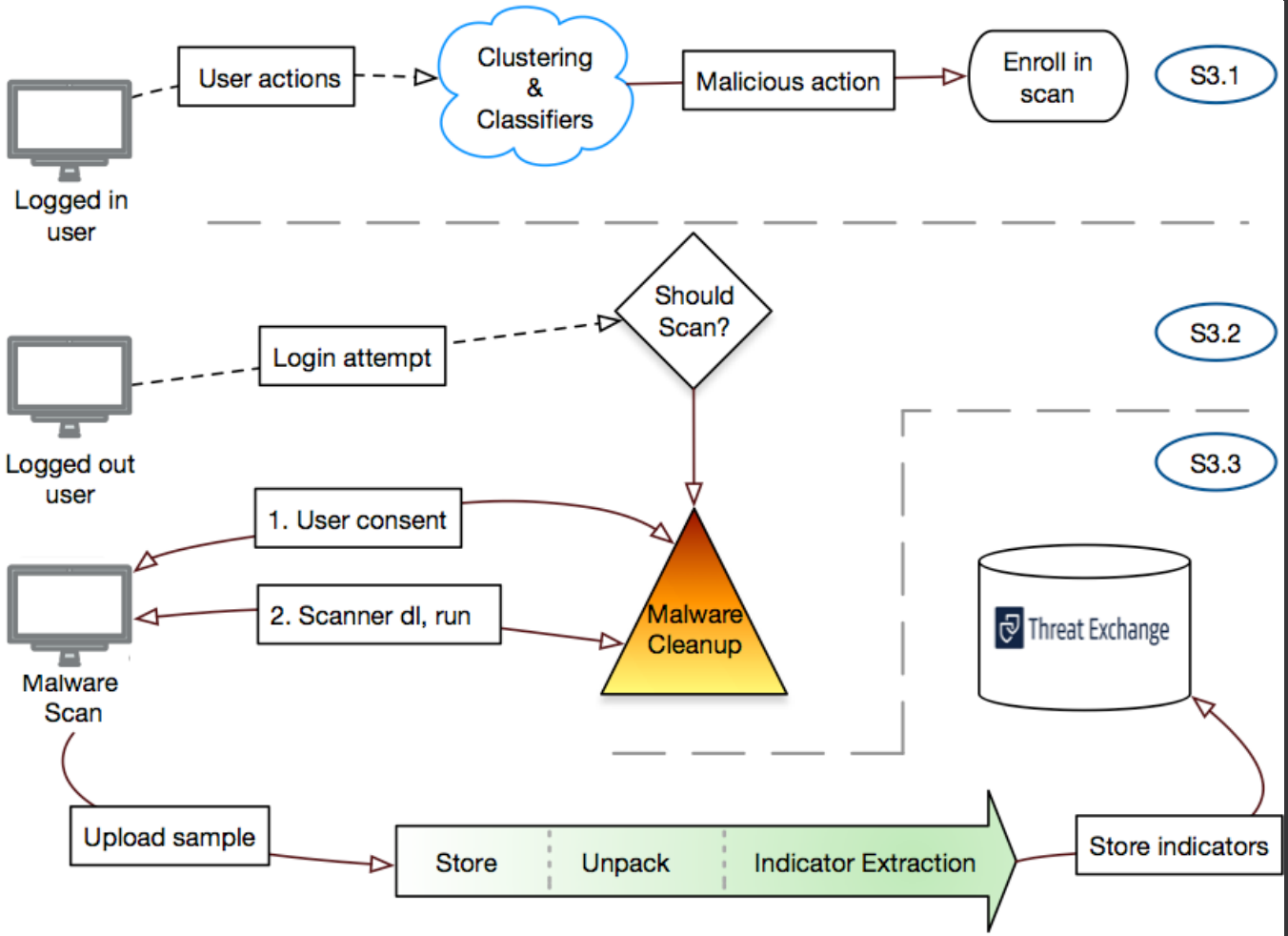
Anti-Malware Scanner

- Facebook's custom scanner is executed on the compromised device following user consent

- Uploads digital fingerprint of extensions to Facebook
  - MD5 hash
- New extensions are uploaded to Facebook
- When MBE are detected they are removed
- Third-party anti-virus scanner executed

System Design

- Detecting compromised user accounts
- Anti-malware scanner
- Static analysis pipeline



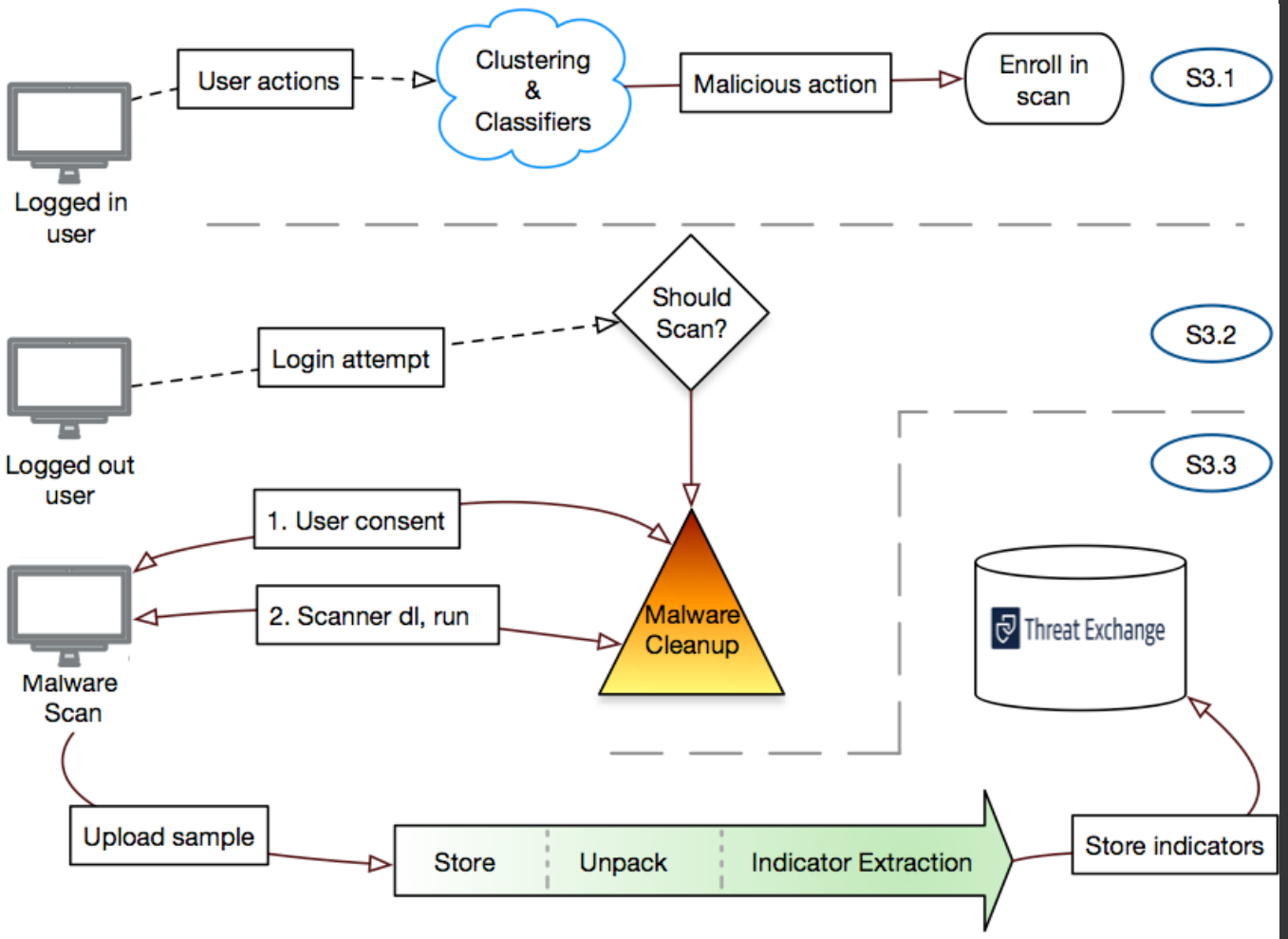
Static Analysis Pipeline

- **Unpacking**
  - Recursively unpack the extension and files
- **Indicator extraction**
  - Deobfuscate, decode, and repair broken URLs
  - Regular expressions extract indicators e.g. URLs, API keys
    - Treating each file as text

- Insight: Extensions collected by Facebook's malware scanner exhibited malicious behavior at the time of collection

System Design

- Detecting compromised user accounts
- Anti-malware scanner
- Static analysis pipeline
- Extension labeling



Indicator Labeling

- `MALICIOUS`
  - Malicious with high-confidence
- `UNKNOWN`
  - Default label for all samples
- `NON_MALICIOUS`
  - Benign samples, or samples from trusted sources

- Labels produced by system that detects compromised accounts

Propagating Indicator Labels

- Apply vetted threat labels to indicators from static analysis
- How do we label extensions?
  - JavaScript contains a `MALICIOUS` URL
  - `MALICIOUS` label propagates to the file
  - `MALICIOUS` label propagates the extension
- Erroneously marked indicators
  - Propagate automatically
  - Rules in place to prevent single indicators from mass-labeling
  - Manual labels overrides automated labeling

Malicious Indicators

	Extension Contents		Extracted Indicators		Scan Sessions	
	JS	HTML	Total #	Malicious (#%)	#	%
Chrome Ext.	67 380	720	66 134	1 559 (2.4%)	718 497	96.9
Firefox Ext.	17 979	16	19 004	609 (3.2%)	257 164	34.7
Total Unique	84 905	733	73 281	1 516 (2.1%)	741 276	100.0

- 6-week measurement period
- Only a small number of all indicators are labeled `MALICIOUS`

Malicious Extensions

	All Extensions		Malicious Extensions	
	#	%	#	% of total
Chrome Ext.	23 376	67.6	1 697	7.3
Firefox Ext.	11 183	32.4	88	0.8
Total Unique	34 559	100.0	1 785	5.2

- A high proportion (5.2%) of malicious extensions is expected as our system targets devices exhibiting malicious behavior
- 422 of 1,697 Chrome MBE were once online Google's Web Store
  - Suggests a high number of MBEs to be side loaded

MBE Detection Rates

- Average 39.5 Chrome MBE/day
- Average 2 Firefox MBE/day

- 92% of new MBE are labeled by a median time of **21 seconds**
- 8% of new MBE are labeled more than one day after collection
  - Detected on 9% of user devices cleaned during the experiment

This result is expected from an indicator-based labeling system as labels can change over time

Known False Positives

- 124 extensions are incorrectly labeled `MALICIOUS`
- 0.8% of all scan sessions removed one or more of these extensions
- Median detection time: 18 days

- This result is expected from an indicator-based labeling system as labels can change over time
- We find the low number of incorrectly labeled MBEs to be an acceptable tradeoff

Evaluating Alternatives

- Was it necessary to create a new system that detects MBE?
- Focus on Chrome extensions
  - Majority of extensions are for Chrome browser
  - Each Chrome extension's Web store presence is checked
  - 2,200/23,376 Chrome extensions *once* on the Chrome Web store

- Facebook labels 422 (19.2%) `MALICIOUS`
- Facebook labels 1,778 (80.8%) `UNKNOWN`

VirusTotal

- Provided with 9,172 unique CRX from authors of Hulk[1]
  - VT was aware of *only* 73 extensions
  - Moreover 5 are labeled `MALICIOUS` by at least 1 anti-virus engine

Facebook cannot use general malware databases to detect MBEs

[1] A. Kapravelos, C. Grier, N. Chachra, C. Kruegel, G. Vigna, and V. Paxson. Hulk: Eliciting Malicious Behavior in Browser Extensions. In *Proc. of USENIX Security*, 2014.

VirusTotal

- Provided with 9,172 unique CRX from authors of Hulk[1]
  - VT was aware of *only* 73 extensions
  - Moreover 5 are labeled `MALICIOUS` by at least 1 anti-virus engine

Facebook cannot use general malware databases to detect MBEs

- Of the 422 MBE identified by Facebook
  - 96 (22.7%) are labeled `MALICIOUS` by one or more anti-virus engine

Facebook cannot rely on anti-malware engines to identify MBEs

[1] A. Kapravelos, C. Grier, N. Chachra, C. Kruegel, G. Vigna, and V. Paxson. Hulk: Eliciting Malicious Behavior in Browser Extensions. In *Proc. of USENIX Security*, 2014.

Google Chrome Web Store

- By the six-week period Google removed 367 of the 2,200
  - 70 MALICIOUS
  - 297 UNKNOWN

Facebook cannot rely on Google to remove all MBE targeting FB

- Does Facebook identify MBEs faster?
  - These 70 MBE have over 1 million installs according the the Web Store
  - Facebook identifies the 70 MBE with a median time of **2.8 days** (67.3 hours) before they are removed from the Web store

Our system successfully reduces the median monetization time of MBE

Take Away

MBE are challenging to address from any single vantage point

- Browser vendors
  - Can restrict extension distribution
  - Have limited insight into abusive extensions in the wild
- Abused sites
  - Directly experience malicious behavior
  - But are not in a position to identify which extensions are implicated

MBE Detection Conclusion

- This system is currently running to protect users of Facebook
- As a result Facebook is able to very quickly detect and remove new MBE at scale

422 Chrome MBE MD5 hashes: https://pastebin.com/nzVGPLnr

- Samples available in VirusTotal and Facebook ThreatExchange

Creating Empirical Bias for Security Decisions

Security Practices

- There are many organizations that provide security practices to users
  - We have a long checklist of security practices
- Users pick the security practices they adopt
- Security outcomes depend on user behaviors
  - Bad behaviors can result in malware infections

How to Evaluate Cyber-Risk?

- Unfortunately security practices are received wisdom
- Most security practices tell you the same thing

- No clear way to answer even simple questions about cyber-risk

Understanding Behaviors and Outcomes

- Understanding what security practices relate to security outcomes has many benefits

OBJECTIVE: Provide data and analyses to empirically evaluate bring **large** portions of cybersecurity

Evaluating Cyber-Risk

- Model user behavior at scale
  - Each model will describe user behavior at on a device
- We focus mainly on recommended security practices
- Correlate user behaviors with know security outcomes at scale

Expected Outcomes

- We hope to evaluate cyber-risk on an empirical basis
  - By correlating security practices and security outcomes
- Can produce a fundamental change in how users protect their devices
- Help security organizations protect their users

Disrupting the Malware Lifecycle

- Infected Devices
  - Detecting Malicious Browser Extensions at Facebook

- User Behaviors and Security Outcomes
  - Putting security outcomes on an empirical basis

Research Interests

- Empirical analysis
  - Disrupting criminal ecosystems
  - Preventing the spread of malware throughout its lifecycle
  - Internet-scale measurements (e.g. DNS abuse)
- Cyber-Risk Modeling
  - Network traffic analysis
- Anomaly Detection
  - Underground market disruption
- Protecting people

Thank you!

Louis F. DeKoven, ldekoven@cs.ucsd.edu

Ph.D. Candidate, UC San Diego

Naval Postgraduate School, Tuesday, October 24th, 2017