MANDIANT®
A FireEye™ Company

Data Stacking

Finding Evil (needle) in the haystack

PRESENTED BY: Deepak Nuli

JUNE 11, 2015

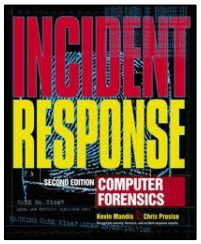
Agenda

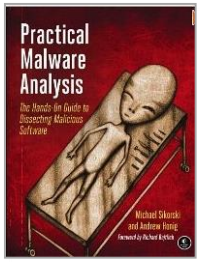
- Background
- Threat Landscape and Investigative Techniques
- Data Stacking
- Case Studies and Examples
- Q&A

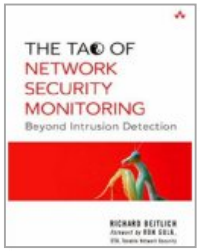
Background

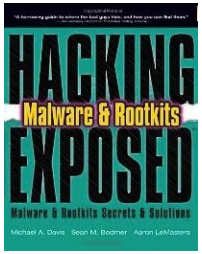
Background on Mandiant

- Founded in 2004 by Kevin Mandia
- Security Consulting
- U.S, Canada, and the World
- Trained the FBI
- Released concrete evidence of hacking by the Chinese PLA in February 2013
- Track hundreds of threat actors
- Mandiant – A FireEye Company









MANDIANT®
A FireEye™ Company

Deepak Nuli

- Senior Consultant
- 6 Years in Information Security
- Social Engineering, Penetration Testing, Incident Response
- Personal Security Blog - *http://hinduhacker.com*
- @hinduhacker

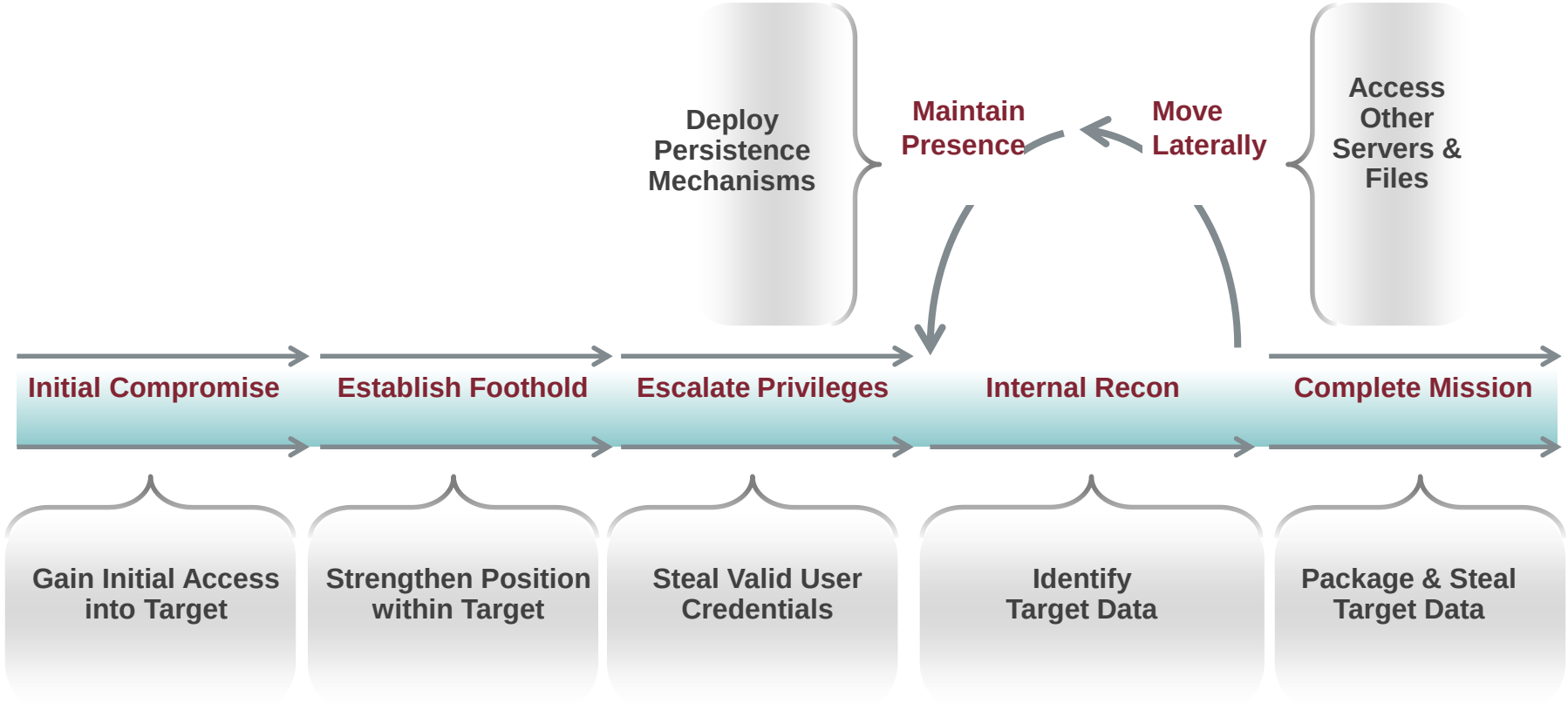
MANDIANT
A FireEye™ Company

Threat Landscape and Investigative Techniques

All Threat Actors Are Not Equal

	Nuisance	Hacktivism	Insiders	Network Attack	Cyber Crime	State Sponsored
						
Objective	**Access & Propagation**	**Defamation, Press & Policy**	**Revenge, Monetary Gain**	**Escalation, Destruction**	**Financial Gain**	**Economic, Political Advantage**
Skill	Low	Low - Med	Med	Med	High	Very High

Anatomy of a Targeted Attack



On average, it takes 205 days for organizations to discover their breach; 30% of organizations self-detected the breach (M-Trends 2015)

MANDIANT®
A FireEye™ Company

Types of Analysis

- Indicators of Compromise (IOCs)
  - Less alert data
  - Low false positives
  - Low/medium false negatives
- Non-signature based - Data Stacking

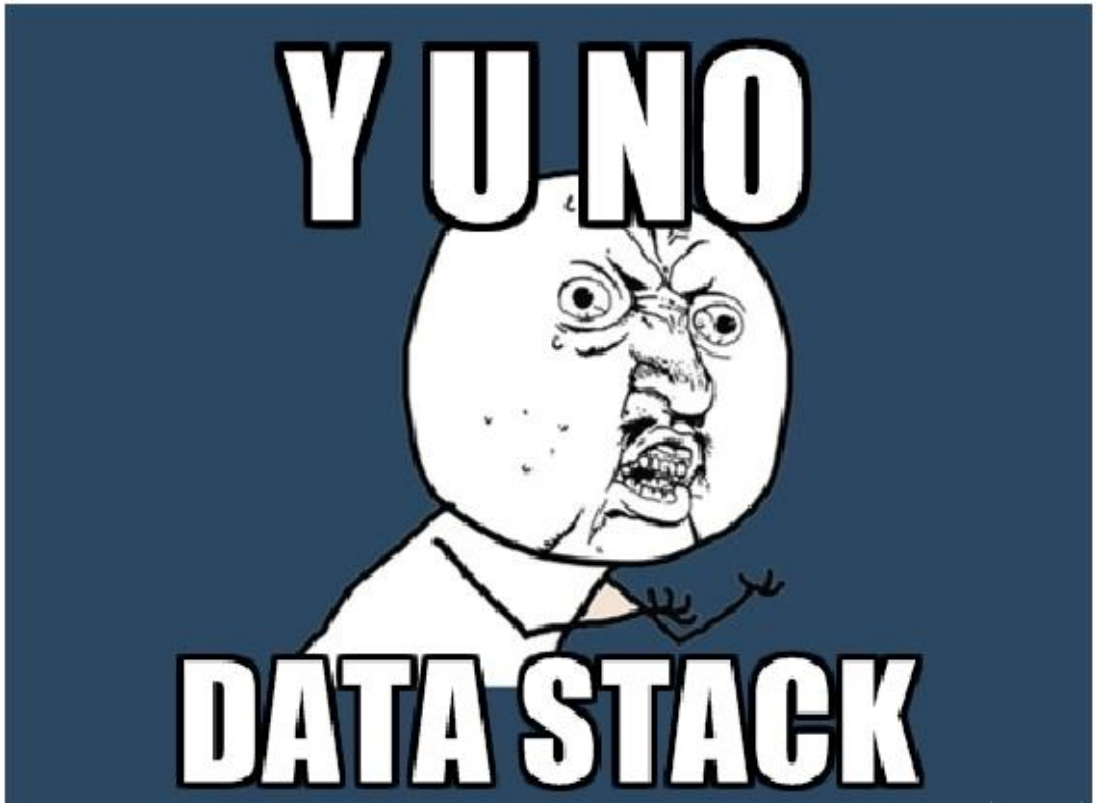
Data Stacking

Data Stacking

- Definition

"Application of frequency analysis to large volumes of similar data in an effort to isolate and identify anomalies" (Mandiant blog)

- Find unknown malware
- Create new IOCs
- Find new compromised hosts



Data Stacking - Methodology

- Step 1: Data Acquisition
  - Large Data Set
- Step 2: Data Filtering
  - Remove known good
- Step 3: Data Grouping
  - Select Attributes and Count Occurrence
- Step 4: Anomaly Detection & Validation
  - Low Occurrences

Data Stacking – Attribute Selection

- High false positives

Data Stacking – Attribute Selection

- High false negatives

Data Stacking Sources

- Current Processes
  - Process Memory
  - Persistent Items – File, Registry, and Services
- Historical Processes
  - Windows Shim Cache Registry
  - Scheduled Tasks
  - Other
    - Altiris Application Metering
    - McAfee Logs
    - etc

Data Stacking - Acquisition Methods

- Commercial Tools
  - IR tools, Application metering, etc
  - Tried and Tested
  - Scale
  - Expensive - $$$ and expertise
- Home Grown
  - Custom Scripts, GPO, WMI, etc
  - Error Prone
  - Can’t scale
  - Less expensive

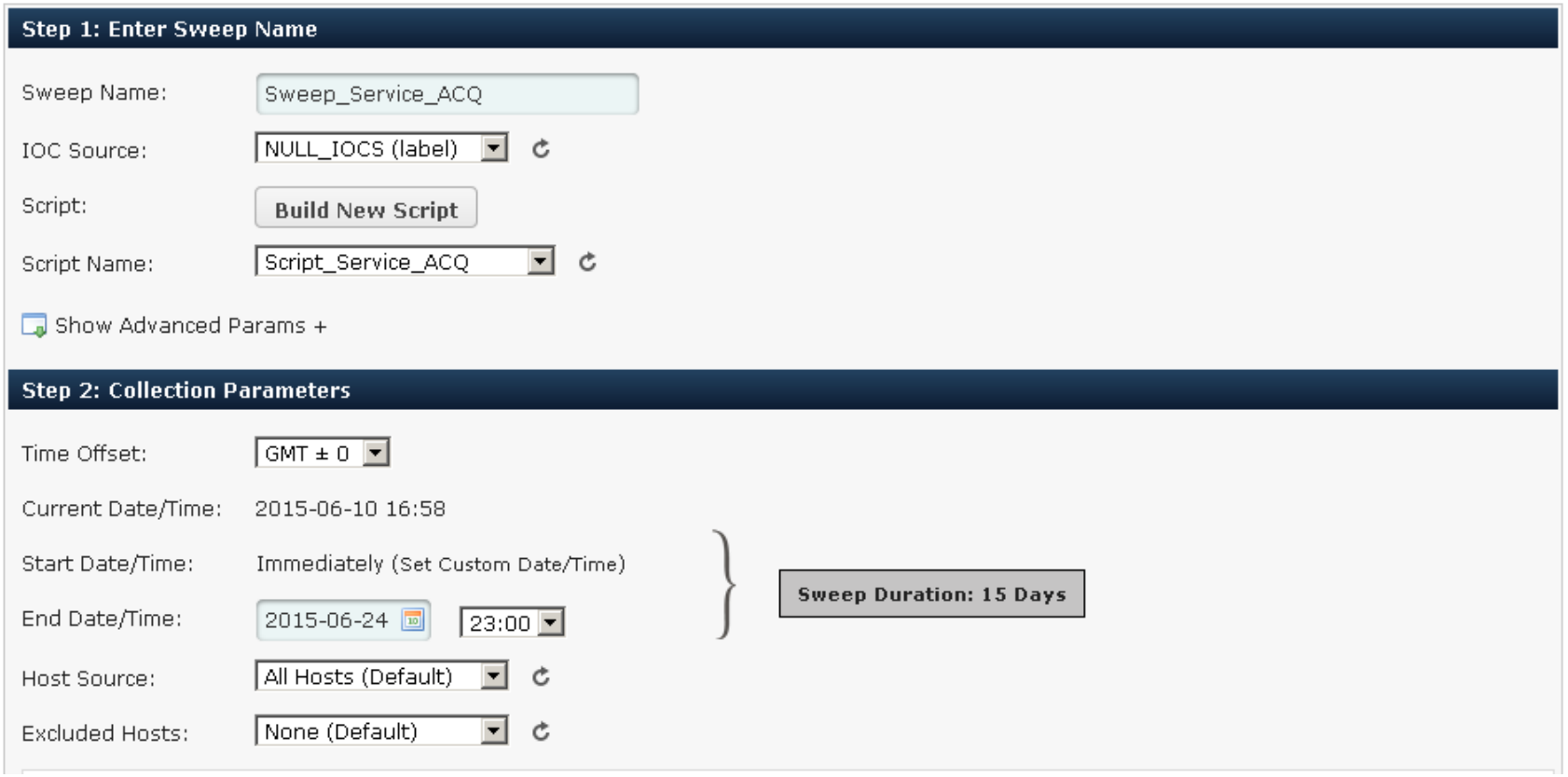
Acquisitions Methods

- MIR
  - Audits

Acquisitions Methods

- MIR

Sweep Configuration



Application Compatibility Cache

Application Compatibility Cache

- Historical File Executions - Registry Entries
- Windows Application Compatibility Database
- Metadata of Files Executed (varies)
  - Name
  - Path
  - Last Modified
  - Size
- Only exe, bat, dll
- Serialization

Application Compatibility Cache

- Step 1: Data Acquisition
  - Total: 20,000+ hosts
  - ~8.5 Million Rows of AppCompat Entries
- Step 2: Data Filtering
  - Program Files\.*
  - Known extensions
  - Known bad paths
  - ~1 Million Unique Rows
- Step 3: Data Grouping
  - File Extensions

Application Compatibility Cache

- Step 4: Anomaly Detection & Validation
  - File Extensions

Count	File Extension
158	dat
79,265	bat
602,343	dll
3,825,229	exe

Count	Full Filename	Description
1	c:\windows\system32\win.dat	WinScanX - Windows password enumeration
8	c:\windows\temp\winlove.bat	Unknown File; Compromised Host

  - New Compromised Hosts: 10 / 40
  - Total Compromised Hosts: ~250

Application Compatibility Cache

- Nitty Gritty – Some examples
  - dd if=input.txt of=output.txt conv=lcase
  - sort input.txt
  - grep –P “\.(exe|dll|bat|dat)” input.txt
  - uniq –c input.txt | awk '{printf("%s,%s\n",$2,$1)}‘
  - sort -t"," -k2 -n

```
root@siftworkstation#grep -v appdata unique_files_with_count|sort -t"," -k2 -n| grep -P "\.dat,"
c:\_[redacted]\ucpsb\cospsb.dat,1
c:\_[redacted]\ucpsb_jdk\cospsb.dat,1
c:\users\administrator\desktop\[redacted]\setupwizard.dat,1
c:\windows\system32\ad.dat,1
c:\windows\system32\g1.dat,1
c:\windows\system32\g2.dat,1
c:\windows\system32\g64.dat,1
c:\windows\system32\win.dat,1
c:\windows\temp\g64.dat,6
```

Service Stacking

Service Stacking

- Service Attributes
  - Name*
  - Descriptive Name*
  - Path*
  - PID
  - Service DLL*
  - Service DLL MD5
  - Service DLL Signature Exists
  - Service DLL Signature Verified

Service Stacking

- Step 1: Data Acquisition
  - Total: 2,266 hosts
  - 1.18 Million Rows of Persistent Data
- Step 2: Data Filtering
  - Service DLLs
  - Remove all Signed and Verified Dlls
- Step 3: Data Grouping
  - Service DLL

Service Stacking

- Step 4: Anomaly Detection & Validation
  - PHOTO backdoor
  - Service name: wuauserv
  - Service DLL: %WINDIR%\system32\msaudXXX.dat

Count	Service DLL	Service DLL Extension
2	C:\windows\system32\<randomname>.dat	dat
20,000+	C:\.*\.dll	dll

Services and Process Memory

Persistent Services and Process Memory

- Relevant Attributes
  - Parent Full File Path
  - Process Full File Path
  - Process MD5*
  - Process Signed*
  - Process Verified*

Persistent Services and Process Memory

- Step 1: Data Acquisition
  - Total: ~5,700 hosts
  - 155,418 Rows
- Step 2: Data Filtering
  - Remove all Signed and Verified Processes/Service DLLs
    - Remaining: 61,306
  - Get unique MD5s
    - Remaining: 48,589
- Step 3: Data Grouping - N/A

Persistent Services and Process Memory

- Step 4: Virus Total API
  - MD5 Search
  - Relevant MD5s: 411

Infection Type	Total Hosts
Adware	72
Generic Trojan	23
Unwanted Programs	7

Summary

- Investigative techniques
- Data Stacking
- Application Compatibility Cache
- Service Stacking
- Services and Process Memory

Q & A

- Questions?
- Say Hii!!
  - deepak.nuli@mandiant.com
  - We Are Hiring!
  - Personal Security Blog -*http://hinduhacker.com*
  - @hinduhacker