Paper Review

Tony Espinoza

am.espinoza@utexas.edu

Nazca: Detecing Malware Distribution in Large-Scale Networks

Nazca: Detecing Malware Distribution in Large-Scale Networks

- A study of how clients in real-world networks download and install malware.
- Nazca is a system that detects infections in large networks.

Nazca does not:

- Operate on individual connections.
- Look at the server hosting the malware.
- Look at the properties of the downloaded program.
- Suffer from coverage gaps in blacklists.

Nazca

- Is a system that aims to detect web requests that are used to download malware binaries.
- Idea is that you need to zoom out to see the whole picture.
- Ideal use case is a large scale:
  - The ISP level.
  - Large enterprise network
  - University

Nazca

Infection process

- Three phases
  - Exploitation
  - Install
  - Control

Antivirus techniques

- What does AV rely on?
- Many antivirus programs rely on signatures
  - What are signatures?
  - What is wrong with signatures?
- What part of the infection process will AV be triggered?

Data set

- One week's worth of traffic
  - From a commercial ISP.
  - What ISP?

Extraction

- Collect pcap files
  - Extract meta-data for connection of interest
  - Connection of interest is?
    - A MIME type not in their blacklist.
  - Re-assemble packets run `file` on the data
  - Try to decompress payload

Extraction

- Record
  - client IP.
  - server IP.
  - URI the client requested.
    - URI = Uniform Resource Indicator.
    - URL = Uniform Resource Locator.
    - URL is subset of URI.
  - User-Agent.
  - Hash of the first k bytes (kilobyte).

Detection

- Use the four techniques to find candidates for malware.
  - Detecting file mutations.
  - Detecting distributed hosting.
  - Detecting dedicated malware hosts.
  - Detecting exploit/download hosts.
- Why these four?

Detect file mutations

- Looks for download records that are:
  - Associated with a single URI.
  - Download more than n different files.

Detecting file mutations

- What about legit sites that do this?
  - They tend to be anti-virus sites.
  - Can white-list easily.

Detecting Distributed hosting and domain fluxing.

- Attackers need their own CDN.
  - Recap, what is a CDN?
  - Why?
- Domain fluxing?
  - The use of many domains to distribute malware.
  - Why would you want to do this?

Find malicious vs benign CDNs

- Six features
  - Domain co-location.
  - Number of unique top-level domain names.
  - Number of matching URI paths.
  - Number of matching file names.
  - Number of URIs per domain.
  - Served file types.

Classifier

- Built a classifier to determine benign or malicious CDN.
- Small data set of manually labeled data
  - How small?
  - What was the split of benign vs malicious?
  - Does it matter?
  - ???

Detection of dedicated malware hosts

- Focus on backend servers that only host malicious binary.
- How does traffic get to the host?
  - Redirection.
- What does this redirection help with?
  - Harder to detect.

Detection of dedicated malware hosts

- Search for IPs that are involved in a single executable download
- Remove all executables that are hosted by other domains
  - Does this leave anything out?
  - Why is this OK?
    - Previous technique handles it.

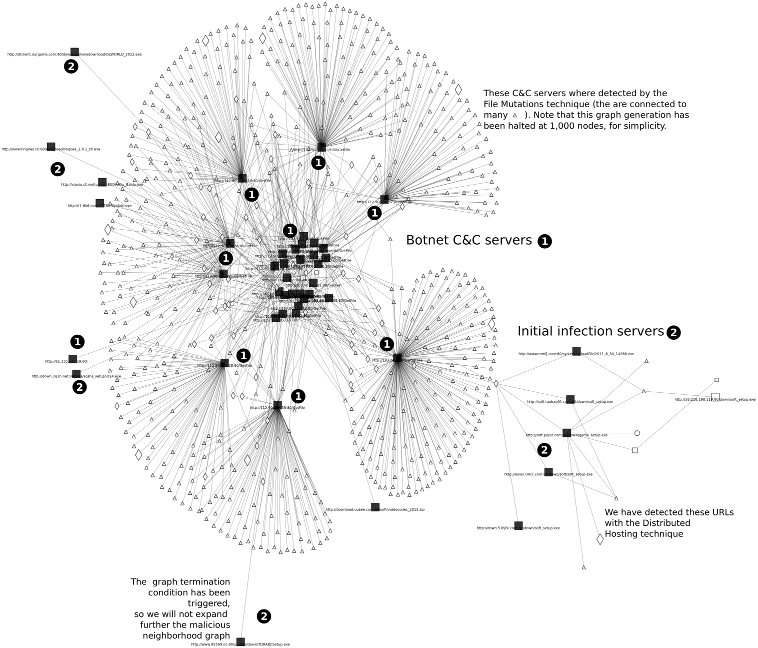
Detection of exploit/Download hosts

- Visit exploit website -> shellcode silently downloads the second step malware binary
- Often times the User-Agent strings do not match.
- This method keeps track of sites that download with a different User-Agent string.
- False positives?
  - Apps in iOS have different user agent strings.
  - Same site multiple browsers.

Detection

- Make neighborhoods of candidates.
- Compute a confidence score.
  - Based on how many other candidates appear in the same graph and how close they are to the input candidate in the graph.

Detection


These C&C servers where detected by the File Mutations technique (the are connected to many △). Note that this graph generation has been halted at 1,000 nodes, for simplicity.
Botnet C&C servers 1
Initial infection servers 2
We have detected these URLs with the Distributed Hosting technique ◇
The graph termination condition has been triggered, so we will not expand further the malicious neighborhood graph
1
2

Graph Generation

- Define malicious neighborhood:
  - The collection of malicious activities related to a suspicious candidate.
  - This is the starting point of the graph.

Graph nodes are:

- IP addresses
- Domain names
- Fully qualified domain names (FQDN)
- URLs
- URL paths
- File names
- Downloaded files (hash value)

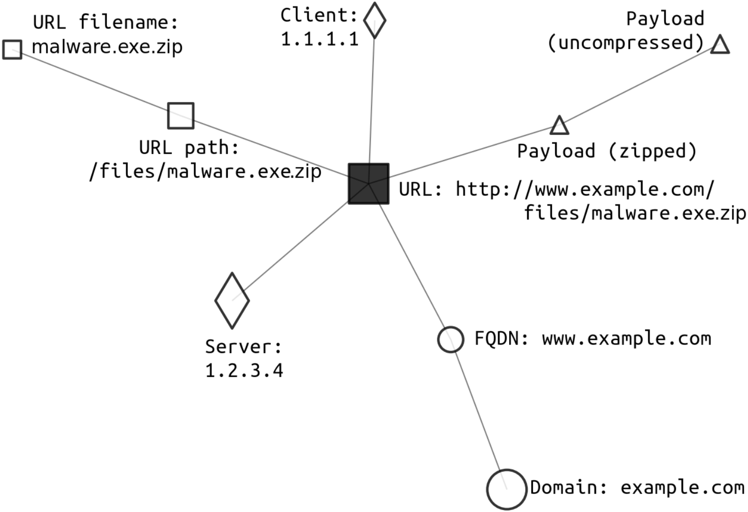
Building a graph

- Incrementally.
- Start with a URL or FQDN.
- Iterate over series of growth operations.
- Limit size to 4,000 nodes.

Growth operations

- Check for relation to entities not in the graph.
  - URLs belonging to a domain or FQDN
  - Files being downloaded form the same URL
  - Domains/FQDNs being hosted on a server
  - URLs having the same path or file name
  - Files being fetched by the same clients

Building a graph



Post-processing

- Remove un-useful parts of the graph.
  - Clients that are leaves in the graph with a single parent.

Graph building rules

- Build towards maximum interest.
  - Towards domains that are not in the graph.
- Only add URLs that are suspicious.
- Avoid adding popular domains $>$ 10% of hosts.

Metric

- Claim an analyst can tell if a graph is benign or malicious.
- Devise a method to check the graph automatically.
- Weight the links.
  - url $\longleftrightarrow$ payload:1
  - url $\longleftrightarrow$ server:1
  - url $\longleftrightarrow$ client:4
  - 2 for all other cases.

Metric

- Lower numbers are better.
- $M_j = \sum_{i=candidates} \frac{1}{shortest\_path(i, j)}$

Evaluation

- Data
  - Training was 2 days
    - April 17 and August 25, 2012
  - Test data was 7 days
    - October 22-28, 2012

Evaluation

- Ground truth
- Where do you get ground truth?
  - Virus total.
- Where did they get the executables?
  - Downloaded them from the URL.
- Issues here?

Evaluation

Technique Type	Malware Train Test	Benign Train Test	Unknown Train Test
File Mutations URLSs	43 8	0 11	16 107
Distributed FQDNs Hosting	45 155	4 17	42 238
Isolated FQDNs Housing	68 145	12 233	47 140
Exploit/ Download Hosts FQDNs	29 16	9 3	28 152

Detection of file mutations

- Need small AV white list
- Trained simple classifier using 5 fold cross validation.
  - Found 12
- 91.1% TP
- 5.8% FP
- TN, FN ???

Detection of distributed Hosting Domain Fluxing

- Leave one out cross validation
- Decision tree classifier
- Learns that malicious distributed hosting infrastructure:
  - Hosts primarily executables
  - Spans across many first-level domains
  - Spans across just 2 or 3 domains

Detection of distributed Hosting Domain Fluxing

Distributed hosting infrastructures	Malicious	Benign	Class Precision
Predicted Malicious	12	5	70.59%
Predicted Benign	0	128	100%
Class recall	100 %	96.24%	

TABLE III. DISTRIBUTED HOSTING: CLASSIFIER PERFORMANCE.

Detection of dedicated malware hosts

- Train single feature classifier
- Uses the User-Agent
- Choose a threshold the equates to 90%
  - 90% of the HTTP requests made with *User-Agent* in the dataset have delivered executables.

Detection step

- Training
  - 77.35% precision 65.77% recall on malicious.
  - 95.7% precision and 97.53 recall on the benign.
- Test
  - 59.81% precision 90.16% recall on malicious.
  - 99.69% precision 98.14% recall on benign.

Detection step

- Malicious class precision, recall?
- Benign class precision, recall?
- What happened to regular old precision, recall, accuracy, f1?
- What about baseline?
- Step back to last table, confusion matrix?

Why HTTP?

- How much traffic was HTTP
- How much traffic was HTTPS
  - HTTPS$<$1%

What about switching?

- Switching would:
  - Be impractical
  - Cause them to self-sign or get a legit cert

How to evade?

- Use HTTPS
- Use custom encryption
- Piggyback
  - Nazca can not detect piggy backing legit services
    - Google
    - Dropbox
- Keep infrastructure small
- Keep churning malware infrastructures
- Disguise as white-listed file type

What did you think?

- Good?
- Bad?
- Questions on?
  - Evaluation.
  - Results.
  - ???

Outside the closed world

The idea of specifying only positive examples and adopting a standing assumption that the rest are negative is called the closed world assumption

Outside the closed world

[The assumption] is not of much practical use in real- life problems because they rarely involve "closed" worlds in which you can be certain that all cases are covered

Outside the closed world claim

"Our main claim is that the task of finding attacks is fundamentally different from other applications, making it significantly harder for the intrusion detection community to employ machine learning effectively."

NIDS

- Network intrusion detection systems (NIDS)
  - Misuse-detection.
  - Anomaly-detection.
- Which is most used in practice?
  - Misuse detectors
  - Why?
    - Anomaly detectors rely on ML

What is anomaly detection good for in network security?

Anomaly detection is likely in fact better suited for finding variations of known attacks, rather than previously unknown malicious activity.

Why not well suited for ML?

- A very high cost of errors.
- Lack of training data.
- A semantic gap between results and their operational interpretation.
- Enormous variability in input data.
- Fundamental difficulties for conducting sound evaluation.

False positives in IDS systems

- What is an acceptable false positive rate?
- What about false negative rate?

Explore other domains

- Ham vs Spam.
- Product recommendation.
- Auto Correct.
- Auto Translate.

Semantic gap

- Identify deviations from the normal profile.
- What does it mean.
  - Abnormal activity vs attacks

Variability in input data

- It is crucial to acknowledge that in networking a high amount of variability occurs regularly
  - Why?
- How to address this?
  - Aggregate data.

Evaluation

- Authors claim:
  - More difficult than building the detector itself
  - Agree, disagree?
- Key issues:
  - Finding Data.
  - Interpreting results.

Key to understand the threat model

- What kind of environment does the system target?
- What do missed attacks cost?
- What skills and resources will attackers have?
- What concern does evasion pose?